



ACOS 6.0.9

Application Delivery Controller Guide

July, 2026

© 2026 A10 Networks, Inc. All rights reserved.

Information in this document is subject to change without notice.

PATENT PROTECTION

A10 Networks, Inc. products are protected by patents in the U.S. and elsewhere. The following website is provided to satisfy the virtual patent marking provisions of various jurisdictions including the virtual patent marking provisions of the America Invents Act. A10 Networks, Inc. products, including all Thunder Series products, are protected by one or more of U.S. patents and patents pending listed at: [a10-virtual-patent-marking](#).

TRADEMARKS

A10 Networks, Inc. trademarks are listed at: [a10-trademarks](#)

CONFIDENTIALITY

This document contains confidential materials proprietary to A10 Networks, Inc. This document and information and ideas herein may not be disclosed, copied, reproduced or distributed to anyone outside A10 Networks, Inc. without prior written consent of A10 Networks, Inc.

DISCLAIMER

This document does not create any express or implied warranty about A10 Networks, Inc. or about its products or services, including but not limited to fitness for a particular use and non-infringement. A10 Networks, Inc. has made reasonable efforts to verify that the information contained herein is accurate, but A10 Networks, Inc. assumes no responsibility for its use. All information is provided "as-is." The product specifications and features described in this publication are based on the latest information available; however, specifications are subject to change without notice, and certain features may not be available upon initial product release. Contact A10 Networks, Inc. for current information regarding its products or services. A10 Networks, Inc. products and services are subject to A10 Networks, Inc. standard terms and conditions.

ENVIRONMENTAL CONSIDERATIONS

Some electronic components may possibly contain dangerous substances. For information on specific component types, please contact the manufacturer of that component. Always consult local authorities for regulations regarding proper disposal of electronic components in your area.

FURTHER INFORMATION

For additional information about A10 products, terms and conditions of delivery, and pricing, contact your nearest A10 Networks, Inc. location, which can be found by visiting www.a10networks.com.

Table of Contents

Getting Started	29
Who Should Read This Guide	29
How This Guide is Organized	29
Deployment	30
Configuration	30
Optimization	30
IPv4 or IPv6 Protocols	31
Web Logging	31
Concepts and Terminology	33
Load Balancing Terminology	34
Real Server	34
Real Server in Virtual Mode	35
Real Server as AAM Authentication Server	37
Known Limitations	38
Virtual Server and Virtual IP (VIP)	38
Service Group	40
Wildcard VIPs, Ports, and Virtual Ports	41
Normal VIP with ACL	42
Source Network Address Translation (NAT)	43
Networking Modes	44
Transparent (Switch) Mode	44
Gateway (Router) Mode	44
Virtual Wire Mode	45
Deployment Modes	45
Inline Mode	45
One-Arm Mode	46

Part I: Application Delivery Controller Deployment	47
Transparent Mode Deployment	48
Transparent Inline Mode	49
CLI Configuration	49
GUI Configuration	51
Transparent Mode in Multinetted Mode	54
Layer 3 IP Source NAT	55
CLI Configuration	55
GUI Configuration	57
Gateway Mode Deployment	61
Gateway Inline Deployment	62
CLI Configuration	63
GUI Configuration	64
Gateway One-Arm Deployment	67
Traffic Flow	68
ACOS Interface Configuration	68
Traffic Blocking Between VLANs	68
Network Address Translation	69
CLI Configuration	69
GUI Configuration	71
Direct Server Return (DSR) SLB Deployment	76
Direct Server Return in Transparent Mode	77
Requirements for DSR in Transparent Mode Deployment	78
DSR Health Checking	78
CLI Configuration	79
GUI Configuration	80
Direct Server Return in Route Mode	82
CLI Configuration	83
GUI Configuration	84
Direct Server Return in Mixed Layer 2/Layer 3 Mode	84

CLI Configuration	86
GUI Configuration	86
Layer 3 Direct Server Return (IP Tunneling)	88
Requirements	88
Methods to Display the ACOS VIP as the Source Address	89
IP-in-IP Tunneling for L3 DSR Routing	89
DSR Health Checking	91
Layer 3 DSR Health Checks	91
Layer 4-7 DSR Health Checks	92
CLI Configuration	92
Route Health Injection Deployment	95
Overview	96
Deployment Examples	97
Site Preference	97
Backup Site	98
Configuration Considerations	99
Basic RHI with BGP	100
Configuring Route Map	101
Consolidated Configuration Example	103
Basic RHI with OSPF	109
Consolidated Configuration Example	109
Advanced RHI with OSPF for VRRP-A	112
Configuring Route map	112
Configuring Prefix list within Route Map	113
Consolidated Configuration Example	114
Network Address Translation	119
Overview	120
SLB Destination NAT	120
SLB Source NAT	121
Connection Reuse	121

Source NAT for Servers in Other Subnets	122
Direct Server Return	124
IP NAT Support for VIPs	125
Priority for Source IP NAT Configurations on Individual Virtual Ports	125
VIP Source NAT	126
Using IP Pool Default Gateways	127
Smart NAT for Virtual Ports	128
Virtual-port TCP Maximum Segment Life	133
Non-standard Port Support	134
CLI Configuration	135
GUI Configuration	135
Dynamic Real Server Creation Using DNS	136
Overview	137
Creating and Maintaining Dynamic Servers	137
Dynamic Server Aging	137
Template Options	138
Server Template Options	138
Server Port Template Options	139
CLI Configuration	140
GUI Configuration	144
Virtual Port Ranges	147
Overview	148
Virtual Port Types	148
Preventing SLB Reset and Ethernet Statistics	149
CLI Configuration	150
GUI Configuration	151
Mapping Virtual IP Addresses and Real Ports	152
Overview	153
Deterministic Mapping	153
Mapping VIP to Real Ports: Supported Virtual Port	155

Mapping VIP to Real Ports: Requirements	155
Virtual IP Creation Based on Subnet	156
CLI Configuration	157
GUI Configuration	158
Part II: Application Delivery Controller Configuration	161
MQTT Load Balancing	162
Overview	162
Deployment Example	163
CLI Configuration	164
FTP Load Balancing	166
Overview	167
CLI Configuration	168
GUI Configuration	168
HTTP Load Balancing	172
Overview	173
Components	174
Service Groups	174
Virtual Server	175
Templates	175
Service Templates	176
Server and Port Templates	177
Health Monitors	177
Configuring an HTTP Template	178
CLI Configuration	179
GUI Configuration	180
Viewing HTTP Statistics	183
HTTP Options for SLB	185
Server and Service Group Selection Options	186
URL Hash Switching	186

Hash Options	187
Configuring Hash Switching with Server Load Awareness	189
Configuring URL Hashing	191
URL / Host Switching	193
Match Options	194
Configuring URL / Host Switching	195
Using URL / Host Switching with Cookie Persistence	197
URL Failover	201
Configuring URL Failover	202
Decompression for HTML Tag Insertion	203
Configuring Content Compression	203
Temporary Compression Disable During High CPU Utilization	206
Strict Transaction Switching	207
Enabling Strict Transaction Switching	207
5xx Retry and Reassignment	208
Non-HTTP Bypass	209
Performance Enhancing Options	210
HTTP Packet Flow Modes	211
High-speed HTTP Content Replacement	212
HTTP Content Format	213
Configuring High-Speed HTTP Content Replacement	213
Content Compression	214
Accept-Encoding Field	214
Compression Level	215
How ACOS Determines Whether to Compress a File	215
CLI Configuration	216
GUI Configuration	218
Brotli Compression	219
CLI Configuration	220
GUI Configuration	221
HTTP Requests Options	222

Header Insertion / Erasure	222
Configuring Header Insertion / Replacement	223
Configuring Header Erasure	226
Client IP Insertion / Replacement	227
Replace Option	228
Configuring Client IP Insertion / Replacement	229
Client-IP Insertion Extensive Support	231
Server Replies Options	231
URL Redirect Rewrite	231
Redirect-Rewrite Rule Matching	231
Configuring URL Redirect Rewrite	232
Inserting HTTP Client Port Numbers in the HTTP Header	233
HTTP Protocol Compliance	233
CLI Configuration	234
Configuring HTTP Header Filter Rules	235
CLI Configuration	235
HTTP/2 Load Balancing	237
CLI Configuration	237
Consolidated Configuration Example	239
HTTP/1.1 to HTTP/2 Communication Support	240
CLI Configuration	242
HTTP/2 SSL Offload without SNAT	245
HTTP Load Balancing to Proxy Servers	247
Overview	248
Configuring HTTP Load Balancing to Proxy Servers	248
Configuring HTTP Headers to Forward to Proxy Servers	249
Responding to Server Selection Failure	250
Overview	250
TCP Reset	250
Gateway Timeout Error	253

CLI Configuration	253
SIP Load Balancing	255
Load Balancing for SIP over UDP	256
Application Layer Gateway Support	256
Configuring Load Balancing for SIP over UDP	257
GUI Configuration	257
CLI Configuration	260
Load Balancing for SIP over TCP/TLS	262
SIP Multiplexing	263
ACOS Requirements	264
Client and Server Requirements	265
Client Keepalive and Server Keepalive	265
ACOS Actions if Selection of a Client or SIP Server Fails	266
SLB Network Address Translation for SIP over TCP / TLS	267
Configuring SIP over TCP / TLS for SIP Multiplexing	268
Disabling Reverse NAT Based on Destination IP Address	274
IP NAT for SIP	276
Financial Information eXchange Load Balancing	278
Overview	279
Tag-based Service Group Selection	279
Client IP Address Insertion	279
FIX Load Balancing Example	280
FIX Load Balancing	281
Configuring FIX Load Balancing	281
GUI Configuration	281
CLI Configuration	284
Viewing FIX Traffic Statistics	287
Server and Port Templates	289
Overview	290
Precedence of Template and Direct Settings	290

Default Server and Port Templates	290
Modifying a Default Template	291
Parameters Configured Through Server and Port Templates	291
Real Server Template Parameters	291
Real Port Template Parameters	292
Virtual Server Template Parameters	292
Virtual Port Template Parameters	292
Configuring and Applying Server and Port Templates	292
Configuring Server and Port Templates	292
Applying a Server or Port Template	294
Binding a Server Template to a Real Server	294
Binding a Server Port Template to a Real Server Port	295
Binding a Virtual Server Template to a Virtual Server	296
Binding a Virtual Server Port Template to a Virtual Port	296
Binding a Server Port Template to a Service Group	297
Port and Server Template Parameters	297
Connection Limiting	298
Connection Limit Parameters	298
Setting a Connection Limit	299
Connection Rate Limiting	300
Connection Rate Limiting Parameters	300
Configuring Connection Rate Limiting	300
Server Flap Dampening	301
Server Flap Dampening Description and Parameters	301
Configuring Flap Dampening	302
Slow-Start	302
Ramp-Up Parameters	302
Behavior When Slow Start Is Also Configured on the Real Server	304
Configuring Slow Start	304
Request Rate Limiting	305
Graceful Shutdown	306

Gratuitous ARPs for Subnet VIPs	307
TCP Reset Option for Session Mismatch	308
Client Port Preservation	309
Health Monitoring	312
Overview	313
Default Health Checks	313
Health-Check Guidelines	314
Health Monitor Natting	315
CLI Configuration	316
Limitations	317
Health Method Timers	317
Health Check Operations	318
Server or Port Is Unresponsive	318
Server or Port Is Responsive	319
Health Method Types	319
Protocol Port Numbers Tested by Health Checks	331
Configuring and Applying a Health Method	332
Configuring an Internal Monitor	332
Import an Externally Configured Monitor	333
Apply a Health Monitor to a Real Server or Real Service Template	334
Apply a Health Monitor to a Real Server or Service Port	335
Health Monitoring Virtual IP Addresses in DSR Deployments	336
Configuring a Layer 3 health method targeted to a Virtual IP Address	337
Globally enable DSR health checking of virtual IP addresses	338
Using Send and Receive Strings in TCP Health Checks	338
Using Customized Header Insertion in SIP Health Check	340
Certificate and Key Support in HTTPS Health Monitors	342
SNI Support for HTTPS Health Monitors	343
TLS Session Resumption Support for HTTPS Health Monitors	344
POST Requests in HTTP/HTTPS Health Monitors	346

Automatic Interval Adjust Based on HTTP Status Code	348
Passive Health-Monitoring Parameters	348
Passive Health-Monitoring Activation	349
Configuring Passive Health-Monitoring Activation	350
Customizing DNS Health Monitors	351
DNS Health Monitoring over TCP	354
Overriding the Target IP Address or Protocol Port Number	355
Override Parameters	357
Configuring the Target IP Address or Protocol Port Number Override	357
Basing a Port's Health Status on Another Port's Health Status	358
Service Group Health Checks	359
Priority of Health Checks	360
Configuring Service Group Health Checks	361
Disable Failed Health Check	363
In-Band Health Monitoring	364
Relationship To Standard Layer 4 Health Monitoring	364
How In-Band Layer 4 Health Monitoring Works	365
Logging and Traps	365
Configuring In-Band Health Monitoring	366
Consecutive Health Checks	367
Maintenance Health Checks	368
On-Demand Health Checks	370
Enabling Strict Retries	370
Globally Changing Health Monitor Parameters	371
Globally Disabling Layer 3 Health Checks	373
Health-Check Rate Limiting	373
Health-check Threshold	374
Health-Check Interval and Timeout	374
Configuring Health-Check Rate Limiting	375
Multi-CPU Support for Health Monitoring	376
Database Health Monitors	377

Database Health Monitor Parameters	377
Required Health Monitor Parameters	377
Optional Health Monitor Parameters	377
Configuring Database Health Monitors	378
Kerberos Health Monitors	378
Compound Health Monitors	380
Compound Health Monitor syntax	380
Compound Health Monitor Considerations	382
Configuring Compound Health Monitors	382
Configurable Response Codes: RADIUS Health Monitors	385
Displaying Health Status	385
Viewing Health of Virtual Servers and Service Ports	386
Viewing Health of Real Servers and Service Ports	387
Viewing Health Monitoring Statistics	388
Using External Health Methods	389
Runtime Environment	390
Supported Runtime Environments and Commands	390
Using External Health Monitors with Root-Shell Access	393
Security and Operational Controls	397
Securing External Health Monitors	397
Restricting Extended Health Monitor Activity	398
Monitoring Extended Health Monitor Activity	399
Reviewing Extended Health Monitor Configuration	399
Configuring External Health Monitors	399
External Health Monitor Script Examples	402
TCL Script Example	403
Perl Script Example	405
Shell Script Example	405
Python Script Example	406
Database Health Monitoring Using a Script	406
Example Script—Microsoft SQL	407

Example Script—MySQL	408
Example Script—PostgreSQL	409
Example Script—Oracle	410
Alternate Real Servers and Ports for Backup	412
Alternate Servers	413
Sequence Numbering of Alternate Servers	413
Re-Activation of Down Servers	413
Configuring Alternate Servers for Backup	414
Alternate Ports	417
Alternate Ports Overview	417
Configuring Alternate Ports	418
Alternate Virtual Ports for Backup	423
Overview	424
Supported Alternate Virtual Port Service Types	424
Virtual Port Switchover Options	424
Configuring Alternate Virtual Ports for Backup	425
Priority Affinity	427
Priority Affinity Example	428
ACOS Behavior – Priority Affinity Disabled	429
ACOS Behavior – Priority Affinity Enabled	430
Priority Affinity Options	430
Priority Affinity Actions	431
Priority Affinity Triggers	431
Configuring Priority Affinity	432
Configuring Basic Priority Affinity	432
Configuring Affinity Priority Levels	433
Configuring ACOS Reactions to Exceeded Connection Limits	434
Source-IP Persistence Override and Reselect	438
Default Source IP Persistence Behavior	439

Behavior With Source-IP Persistence Override and Reselect	439
Simplified Source IP Persistence Override Example	439
Source IP Persistence Override Use Case Scenario	440
Configuring Source IP Persistence Override	440
Policy-Based (PBSLB) Templates	442
PBSLB Options	443
Forward Policy Options	444
Configuring Forward Policy in PBSLB Template	447
CLI Configuration - Example 1	447
CLI Configuration - Example 2	448
Configuring and Binding Forward Proxy Objects	450
CLI Configuration	450
Viewing Forward Policy Statistics	451
Show Command Example	452
Binding Class-List to PBSLB Template	452
Binding PBLBSB Templates to Service Groups	454
Binding Policy Template to Virtual Ports	456
Viewing PBSLB Statistics	460
Scan-All-Members Option (Persistence)	461
Overview	462
Configuring Scan-All-Members	464
Quality of Service Marking for TCP Traffic	467
Overview	468
Configuring Quality of Service for TCP	469
GUI Configuration	469
ADC Multi-PU Configuration	470
Traffic Flow	470
Key Considerations	472
CLI Configuration	472

Consolidated Configuration Example	477
Limitation	481
Part III: Application Delivery Controller Optimization	482
SSL Certificate Management and Options	483
SSL Certificate Management	485
CA Certificate Versus SSL Certificate	486
Process	486
Certificate Chain	488
Certificate Warning from Client Browser	489
CA-Signed and Self-Signed Certificates	490
SSL Templates	491
Client-SSL Template Configuration and Usage Guidelines	491
Server-SSL Template Configuration and Usage Guidelines	495
Cipher Template Configuration and Usage Guidelines	497
Key Considerations	497
TLS Server Name Indication (SNI) Support	497
Default Certificate and Key	498
SSL Server Certificate Authorization Validation	498
SNI Extension Support	499
Partition Support	500
Configuring TLS Server Name Indication	500
TLS SNI Support on vThunder	501
CAs and CSRs Management	503
Importing a Certificate and Key	504
Importing Individual Files	504
SSL Certificate and Key Files Bulk Import	505
Generating a Certificate Signing Request (CSR)	506
Generating a Self-Signed Certificate and Key	507
Generating an SSL Certificate – Private Key File with a CSR	509
Installing Certificates	513

Requesting and Installing a CA-Signed Certificate	513
Installing a Self-Signed Certificate	515
Implementing Certificates to SSL Templates	516
Binding Server-SSL Templates to Individual Real Ports	517
Creating Multiple CA Certificate in Server-SSL Templates	519
Multiple Certificates in Single File – Preparing the File	519
Configuring Email Notification for SSL Certificate Expiration	521
Converting Certificates and CRLs to PEM Format	522
Importing a Certificate Revocation List (CRL)	524
Exporting Certificates, Keys, and CRLs	524
Importing a CA Cert and Private Key	526
Configuring Forward Proxy Alternate Signing Cert	526
Deleting Certificate Files	527
SSL Certificate Notification via System Log Warnings	528
Configuring Simple Certificate Enrollment Protocol (SCEP) Certificates	528
Key Considerations	528
Enrollment and Renewal Process	529
Configuring SCEP Certificate	530
Configuration Example	530
Configuring Automatic Certificate Management Environment (ACME) Certificates	534
Enrollment and Renewal Process	535
Configuring ACME Certificate	537
Configuring External Account Binding	538
Configuration Examples	538
SSL Offload and SSL Proxy	540
SSL Offload	541
Overview	541
Configuring SSL offload	541
SSL Proxy	546
Overview	546

Configuring SSL Proxy	547
Overview	549
Configuring SSL Ciphers	550
Configuration Examples	551
Secure FTP Proxy	553
Overview	554
FTP Proxy Traffic Flow	555
Explicit Passive FTP Proxy Session Example	555
Configuring SSL Offload for Secure FTPS Traffic	557
ALG Protocol FWLB Support for FTP and SIP	561
Overview	562
FTP Support	562
SIP Support	563
Topologies	564
FTP Network Diagram	565
SIP Network Diagram	566
Persistent Sessions	567
FTP Persistent Sessions	567
External-facing ACOS Device (client-side)	567
Internal-facing ACOS Device (server-side)	568
SIP Persistent Sessions	569
Internal-facing ACOS Device (server-side)	569
External-facing ACOS Device (client-side)	569
Configuring ALG FWLB	570
Wildcard VIP	588
ACLs for Identifying Valid Traffic	588
Wildcard Protocol Ports on the Wildcard	589
Server and Service-group Requirements	590
Promiscuous Mode	591
Session Persistence for FTP and SIP	591

FTP-Specific Configurations	591
SIP-Specific Configuration	592
Health Monitoring for FWLB Paths	592
DNS Optimization	594
Overview	595
Global DNS Optimization	595
DNS Optimization per VIP	597
Class-List Parameters	598
DNS Template Parameters	599
DNS Flag Day Compliance	606
Configuring DNS Caching	607
Overview	608
Configure a Class List	608
Configure the GLIDs	609
Configure a DNS Template	610
Enable DNS Caching on a VIP	611
Display and Clear DNS Caching Information	612
Using the CLI	612
Using the GUI	612
Display DNS Cache Hits and DNS Query Rate	613
Limitation	613
Synchronize Global DNS Cache	614
Configure Global DNS Cache	615
Enable Persistent Storage for Global DNS Cache	618
Configure DNS Cookie Cache Policy	620
Using the CLI	621
Configure Negative Caching of DNS Response	621
DNS Cache Logging	625
Enable Hit Count of DNS Cache Entries	625
DNS Logging	626

Overview	626
DNS Logging Types	627
DNS Logging Examples	634
CLI Configuration	641
GUI Configuration	642
DNS Queries per Second Logging	643
Authenticate DNS Requests	646
Configuration Examples – DNS Optimization	647
Load Balancing with the “DNSSEC OK” (DO) Bit	657
DNS Recursive Resolver	659
Supported Features	661
Limitation	661
DNS Recursive Resolver Parameters	661
CLI Configuration	664
Show Command	665
DNSSEC Validation	665
Validation Workflow	665
Limitations	666
Supported Algorithms or Digest Types	666
Recursive DNS Gateway Health Monitoring	667
CLI Configuration	668
Show Command	668
DNS Over HTTP/HTTPS (DoH)	669
Known Issues and Limitations	670
CLI Configuration	671
GUI Configuration	672
DNS Filtering Using Web Categories	673
Overview	674
Workflow	674
DNS Filtering Actions	675
Additional Options	676

DNS Filtering Outcomes	677
Example Configurations	677
CLI Configuration	678
Show Commands	681
RAM Caching	682
Overview	683
RFC 2616 Support	683
If-Modified-Since Header Support	684
Support for no-cache and max-age=0 Cache-Control Headers	684
Age and Via Headers into Cached Responses	685
ACOS RAM Cache Behavior	685
Dynamic Caching	687
Host Verification	687
Configuring RAM Caching	688
Transparent Cache Switching	694
Overview	695
TCS Granularity	696
Using Multiple Cache Servers	696
Application Templates	696
Spoofing Caches	698
High Availability Support	698
Configuring Transparent Cache Switching	698
Configuring Layer 4 TCS	699
Configuring Layer 7 TCS	702
Service Type HTTP Without URL Switching Rules	702
Service Type HTTP with URL Switching Rules	704
Optimizing TCS with Multiple Cache Servers	707
Enabling Support for Cache Spoofing	709
Configuring TCS: VRRP-A High Availability	710
TCS: IPv4 VRRP-A High Availability Layer 3 Inline Mode	710

TCS : IPv6 VRRP-A High Availability Layer 3 Inline Mode	719
Configuring TCS for FTP	727
Configuring Bandwidth Limits Per Server or Port	730
Configuring Destination Hash-Based TCS	734
Diameter AAA Load Balancing	736
Overview	737
Source NAT	738
Load-Balancing	738
Health Monitoring	739
Application Templates	739
Accounting Message Duplication	740
Parameters	740
Diameter Attribute-Value Pairs	740
Load Balancing for Specific Message Codes	742
Timers	743
Accounting-Request Message Duplication	743
Configuration	744
RADIUS Message Load Balancing	749
Overview	750
Deployment	750
Server Traffic Flow	751
Stateless Load Balancing Protocol Ports Supported	752
RADIUS Session Aging	753
Configuration	753
SNMP-Based Load Balancing	758
Overview	759
Requirements	759
Weight-based Load Balancing	760
Health Monitoring	761
Configuration	762

Advanced Traffic Replication	765
Overview	766
Topology	766
Dataflow	767
Packet Header Changes	769
Modes	770
Use Case Scenarios	771
Implementation Details	773
Configuration	773
Outbound Next Hop Load Distributor	775
Next Hop Load Distributor Overview	776
Load Balancing Methods	777
Requirements	777
Configuration	778
Configuration Based on Latency	781
Configure NHLD (CLI)	783
HTTP Proxy	785
Overview	786
Configuring HTTP Proxy	789
Resources Parameters	789
Basic Transparent Proxy Configuration	792
CLI Configuration	793
Basic Explicit Proxy Configuration	796
CLI Configuration	797
GUI Configuration	800
Advanced Explicit Proxy Configuration	805
CLI Configuration	806
GUI Configuration	811
Explicit Proxy URL Filtering	818
CLI Configuration	819

GUI Configuration	820
Transparent Proxy URL Filtering	822
URL Filtering Based on SNI	822
URL Filtering Based on SAN	829
Resolve Unknown URLs	837
CLI Configuration	838
GUI Configuration	838
Customizable Response Codes on Blocking HTTP Requests	839
CLI Configuration	840
GUI Configuration	841
Logging HTTP Proxy	842
Syslog	842
CEF	844
Displaying HTTP Statistics	845
Displaying Statistics for Forward Policy	847
Displaying or Clearing Learned Cache Entries	848
Displaying HTTP Explicit Proxy Category-Lists Counters	848
Proxy Chaining SLB	850
Configuring Proxy Chaining	851
Explicit Proxy Chaining	851
HTTP/2 Explicit Proxy Chaining	853
Explicit Proxy with AAM Policy	855
CLI Configuration	856
Explicit Proxy with Virtual Wire Interface	862
Explicit Proxy with Dual-Stack Forwarding Support	863
Deployment	864
Traffic Workflow	865
CLI Configuration	866
Explicit Proxy Without SSLi	870
Terminating SSLi Session Silently	871
Transparent Proxy with Normal VIPs Associated to ACL	873

Overview	873
Key Considerations	873
Limitations	874
Basic Configuration	874
Consolidated Configuration Example	876
Explicit FTP Proxy	878
Overview	879
Features	879
Known Limitations	880
Configuring Explicit FTP Proxy	880
Configuration Resources for Explicit FTP Proxy	880
FTP Proxy Configuration Procedures	884
Basic FTP Proxy Configuration Procedure	884
Explicit FTP Proxy Procedure	884
FTP Proxy Configuration Examples	884
Displaying and Clearing SLB FTP Proxy Statistics	891
Redirection of Traffic to ICAP Servers	894
Overview	895
Configuring ICAP	895
ICAP Deployment Models	896
Basic ICAP Deployment	896
ICAP with SSL Insight Deployment (Optional)	896
ICAP Logging	896
CLI Configuration	897
Part IV: ACOS Performance Tuning	899
Stateless Server Load Balancing	900
Overview	901
Methods	901
Configuration	902

Stateful Hash-Based Server Load Balancing	904
Overview	905
Methods	905
Configuration	905
Auto-Switch to Stateless SLB Based on Traffic	907
Overview	908
Auto-Switch Options	908
Configuration	909
TCP-Proxy	911
Configuration	912
Part V: Protocol Load Balancing	914
IPv4 Load Balancing	915
Overview	916
NAT Support	918
Template Support	918
Direct Server Return	919
Configuring IPv4 Load Balancing	919
IPv6 Load Balancing	923
Overview	924
CLI Configuration	924
Layer 4 TCP/UDP Load Balancing	928
Overview	929
Layer 4 Load Balancing Data Structures	930
Service Groups	930
Virtual Server	930
Templates	930
Idle Timeouts	930
Configurable TCP Half-Open Timer	931
Source-IP Persistence	931

Layer 4 Load Balancing Health Monitors	932
Configuring Layer 4 Load Balancing	932
TCP Fast Open	937
TCP Fast Open Overview	937
Configuring TCP Fast Open	937
SLB Protocol Translation	939
Overview	940
Requirement	941
Configuring SLB Protocol Translation	941
Proxy Protocol	949
CLI Configuration	950
Part VI: Logging	952
Web Logging for HTTP and RAM Caching	953
Overview	954
Log Message Format	954
Configuring Web Logging	954
Web Log String Customization	960
W3C Log Message Format	960
W3C Format Characters	960
W3C Control Characters	962
W3C Format Consideration	963
Configuring Web Logging Format	963
Real-Time Logging for Failed Server Selection	965
Overview	966
CLI Configuration	966
GUI Configuration	968
Glossary	969

Getting Started

This section explains how to configure and manage the Application Delivery Controller (ADC) in the ACOS ecosystem.

Who Should Read This Guide

This guide is intended for experienced IT managers and network administrators who are responsible for configuring and maintaining ADC Thunder devices.

How This Guide is Organized

The Application Delivery Controller allows you to share and distribute the load among multiple servers, thus improving throughput and performance beyond the capabilities of any single server. It also provides fault tolerance and redundancy. Distributing the load among multiple devices enables more network reliability if one server becomes unavailable.

This guide provides deployment examples, configuration steps, and information on how to optimize and tune network performance.

The following sections are included:

- [Part I: Application Delivery Controller Deployment](#)
- [Part II: Application Delivery Controller Configuration](#)
- [Part III: Application Delivery Controller Optimization](#)
- [Part IV: ACOS Performance Tuning](#)
- [Part V: Protocol Load Balancing](#)
- [Part VI: Logging](#)

Deployment

The section on [Part I: Application Delivery Controller Deployment](#) covers two types of in-line deployment modes for ACOS devices: transparent (Layer 2) and gateway (Layer 3). In transparent mode, ACOS devices and real servers are in the same subnet. In gateway mode, the web gateway is placed directly in the path of all network traffic going to and from the Internet.

ACOS secures database requests by applying SSL and TLS protocols on the front- and back-end servers to mask sensitive user credentials and validate client credentials against username-password pairs. In addition, the ACOS can screen requests against aFlex scripts to parse SQL queries and intelligently direct queries among servers.

For more information on application delivery controller concepts (including example topologies deployment), see [Part I: Application Delivery Controller Deployment](#).

Configuration

This section explains how to configure the following:

- FTP Services, optimize the download experience for clients by balancing FTP traffic across servers in a server farm
- HTTP Components, such as service groups, virtual server, templates, and health monitors
- TCP Resets
- SIP
- Financial Information eXchange
- Peer-to-Peer

For more information on application delivery controller configuration, see [Part II: Application Delivery Controller Configuration](#).

Optimization

Stateless ADC conserves system resources by operating without session table entries on the ACOS device. Session table entries contain information about sessions,

including client, VIP, and real server IP addresses and protocol ports. Session table entries also may contain additional state information for specific features.

If the ACOS device is running short on sessions, you can use stateless ADC where applicable to make more sessions available for traffic that requires session table entries.

Stateless ADC is valid for the following types of traffic:

- Traffic with very short-lived sessions, such as DNS
- Layer 2 Direct Server Return (DSR) traffic
- Other types of traffic that do not require features that use session-table entries.

For more information on application delivery controller optimization, see [Part III: Application Delivery Controller Optimization](#).

IPv4 or IPv6 Protocols

IP protocol load balancing enables you to easily load balance traffic based solely on whether the traffic is TCP, UDP, or others such as ICMP (not UDP or TCP), without the need to specify the protocol port numbers to be load balanced.

You can combine IP protocol load balancing with other load balancing configurations. For example, you can use IP protocol load balancing along with HTTP load balancing. In this case, HTTP traffic to the VIP HTTP port number is load balanced separately from traffic to other port numbers.

For more information on protocols that support application delivery controller, see [Part V: Protocol Load Balancing](#).

Web Logging

ACOS supports web logging for HTTP virtual ports, which can be used with HTTP load balancing or RAM caching. Web logging for load-balanced HTTP servers provides data about client access to servers. Web logging for RAM caching provides information about client access to content cached on the ACOS device. Web logging to external log servers is supported over TCP and UDP. Logging over TCP is applicable to web logging for HTTP virtual ports.

For more information on web logging for application delivery controller, see [Part VI: Logging](#).

Concepts and Terminology

This section provides an overview of server load balancing concepts, using example topologies to illustrate how an ACOS device can be deployed and covers these topics.

The following topics are covered:

Load Balancing Terminology	34
Networking Modes	44
Deployment Modes	45

Load Balancing Terminology

The primary goals of server load balancing are to:

- Share and distribute the load among multiple servers, thus improving throughput and performance beyond the capabilities of any single server.
- Provide fault tolerance and redundancy. Distributing the load among multiple devices enables more network reliability if one server becomes unavailable.

This section defines common load balancing terms that are used throughout this and other documents.

The following topics are covered:

Real Server	34
Real Server in Virtual Mode	35
Real Server as AAM Authentication Server	37
Virtual Server and Virtual IP (VIP)	38
Service Group	40
Wildcard VIPs, Ports, and Virtual Ports	41
Normal VIP with ACL	42
Source Network Address Translation (NAT)	43

Real Server

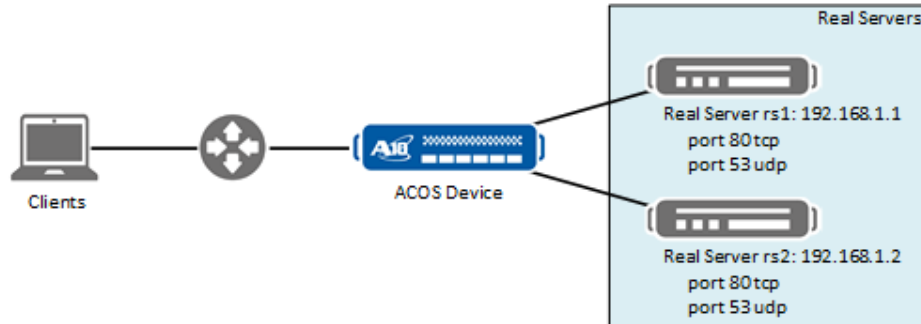
A real server is the physical servers (either individual servers, or servers in a server farm) connected to the ACOS device, or to another switch in the network. [Figure 1](#) displays real servers in a basic SLB deployment.

To configure a real server, use the `slb server` command from the CLI. The minimum configuration for a real server includes the following:

- Name (for example, rs1)
- IP address (for example, 192.168.1.1) or DNS name

- Ports (for example, port 80 for TCP)

Figure 1 : Real Servers



The following is an example of a real server configuration.

```
ACOS(config)# slb server rs1 192.168.1.1
ACOS(config-real server)# port 80 tcp
ACOS(config-real server-node port)#
```

The following is an example of removing a real server configuration.

```
ACOS(config)# no slb server
```

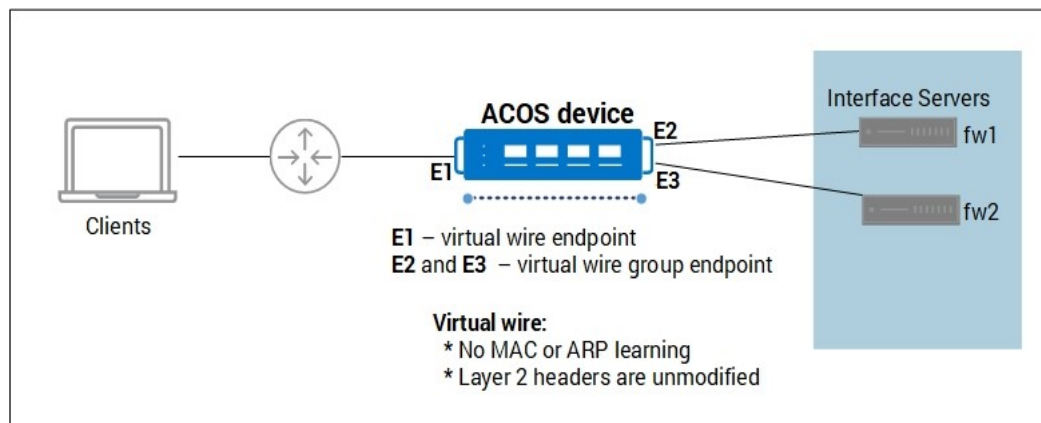
The deletion of the SLB objects is dependent on the global object deletion configuration. For more information, see *Disabling the Deletion of Referenced Objects* section in the [System Configuration and Administration Guide](#).

Real Server in Virtual Mode

A real server is referenced as an "interface server" in a virtual wire configuration for load balancing. The virtual wire endpoint ([Virtual Wire Mode](#)) for the interface servers can be configured as an Ethernet port (for a standalone server), trunk, or virtual wire Ethernet port group for load balancing multiple servers.

The virtual wire endpoints are IP-Less and do not execute health-checks to the target server. The state of the interface server is the same as the associated Ethernet interface and supports both IPv4 and IPv6 traffic.

Figure 2 : Virtual wire with Interface servers



Based on the above figure, the following example configures a virtual wire using an Ethernet endpoint for the client side and a virtual wire Ethernet port group endpoint for multiple ports used for firewall load balancing (FWLB).

Virtual Wire configuration

```
ACOS(config)# interface ethernet 1
ACOS(config-if:ethernet:1)# virtual-wire
ACOS(config-if:ethernet:1)# ip allow-promiscuous-vip
ACOS(config-if:ethernet:1)# exit
ACOS(config)# interface ethernet 2
ACOS(config-if:ethernet:2)# virtual-wire
ACOS(config-if:ethernet:2)# exit
ACOS(config)# interface ethernet 3
ACOS(config-if:ethernet:3)# virtual-wire
ACOS(config-if:ethernet:3)# exit
ACOS(config)# virtual-wire-ethernet-group 1
ACOS(config-virtual-group)# ethernet 2 to 3
ACOS(config-virtual-group)# virtual-wire 1
ACOS(config-if:ethernet:3)# ethernet 1 virtual-wire-ethernet-group 1
ACOS(config-if:ethernet:3)# exit
```

For detailed configuration commands, refer *Networking Configuration Guide*.

FWLB configuration in a virtual wire deployment

The interface servers (firewalls) are configured using the command, `ethernet <num>` along with the service group and wildcard VIP. The wildcard VIP is used to accept all destination IP addresses with DST port 80 from the clients. The command `ip allow-promiscuous-vip` is used on the client side interface to process client side traffic.

```
ACOS(config)# virtual-wire-ethernet-group 5
ACOS(config-virtual-group)# ethernet 2 to 3
ACOS(config-virtual-group)# exit

ACOS(config)# virtual-wire-ethernet-group 10
ACOS(config-virtual-group)# ethernet 4 to 5
ACOS(config-virtual-group)# exit

ACOS(config)# virtual-wire 1
ACOS(config-virtual-wire)# ethernet 1 virtual-wire-ethernet-group 5
ACOS(config-virtual-wire)# exit

ACOS(config)# virtual-wire 2
ACOS(config-virtual-wire)# virtual-wire-ethernet-group 10 ethernet 6
ACOS(config-virtual-wire)# exit

ACOS(config)# slb server s2 ethernet 2
ACOS(config-server)# port 443 tcp
ACOS(config-server)# port 8080 tcp
ACOS(config-server)# exit

ACOS(config)# slb server s3 ethernet 3
ACOS(config-server)# port 443 tcp
ACOS(config-server)# port 8080 tcp
ACOS(config-server)# exit
```

Real Server as AAM Authentication Server

In case the user is using Application Access Management (AAM) to optimize the Authentication, Authorization, and Accounting (AAA) for the client-server traffic then

the real server is created after binding the authentication server to the authentication service group.

The SLB real server can reuse the same server IP as the authentication server and functions as AAM.

Known Limitations

- The real server is reused between the authentication server and the SLB server.
- The real server shared the statistics between the authentication server and the SLB server. However, the service-group does not share the statistics between the authentication server and the SLB server.
- The real server related health-check must be applied to the authentication server only, reused by the SLB server.
- The users can still configure other options like weight, service-principal-name, etc. in the SLB server. However, these may affect the behavior of the authentication server.

For more information on configuring the authentication server, see *Application Access Management* guide.

The following is an example of a the SLB real server reusing the authentication server configuration.

```
ACOS(config)# slb server 192.168.90.136 use-aam-server <----  
authentication server for LDAP is reused here.  
ACOS(config-real server)# port 389 tcp <----- LDAP service-group bound  
with authentication server is reused here.  
ACOS(config-real server-node port)#
```

For more information on CLI configuration, see *Command Line Reference for ADC* guide.

Virtual Server and Virtual IP (VIP)

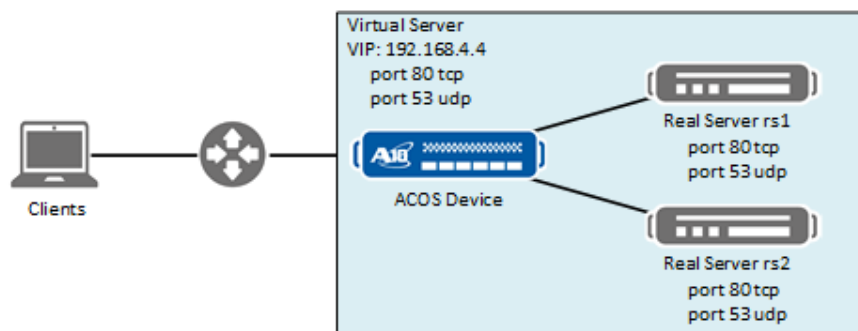
A virtual server is the combination of the real servers and ACOS device, which together appear as a single server to the client

A virtual IP (VIP) is the virtual server IP address. Clients use this IP address to access the virtual server and is configured on the ACOS device. To the client, the VIP represents the individual real server or server farm.

Virtual servers and VIPs are configured by using the `slb virtual-server` command from the CLI. The minimum configuration for a virtual server includes the following:

- Name (for example, vs1)
- IP address (for example, 192.168.4.4)
- Ports (for example, port 80 for TCP)

Figure 3 : Virtual Server and VIP



These commands are an example of this minimum configuration:

```
ACOS(config)# slb virtual-server vs1 192.168.4.4
ACOS(config-slb vserver)# port 80 tcp
ACOS(config-slb vserver-vport)#
```

The following is an example of removing a virtual-server configuration:

```
ACOS(config)# no slb virtual-server
```

The deletion of the SLB objects is dependent on the global object deletion configuration. For more information, see *Disabling the Deletion of Referenced Objects* section in the [System Configuration and Administration Guide](#).

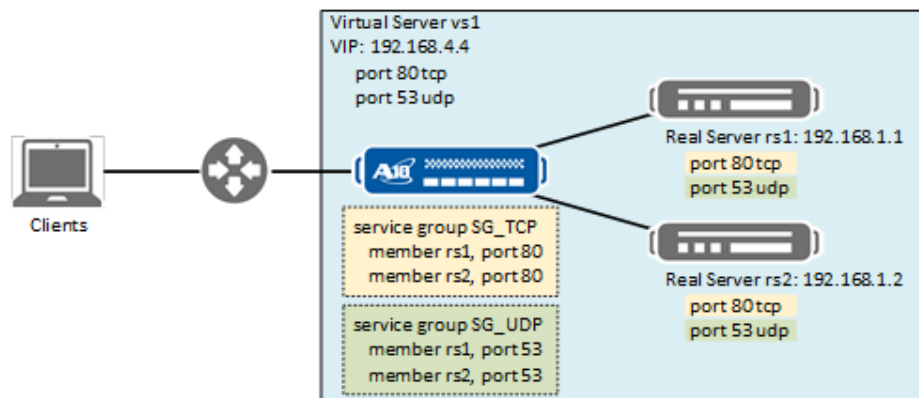
Service Group

A service group is a group of servers that fulfill a service. Service groups are where load balancing algorithms are applied. [Figure 4](#) displays a topology that utilizes two service groups.

Service groups are configured by using the `slb service-group` command from the CLI. The minimum configuration for a service group includes the following:

- Name (for example, sg1)
- Type (for example, TCP)
- Load balancing algorithm (for example, round robin)
- At least one member real server and port (for example, rs1 and port 80)

Figure 4 : Service Groups



Below is an example of this minimum configuration. First, configure two real servers “rs1” and “rs2”. The servers will use port 80 for TCP and port 53 for UDP:

```
ACOS(config)# slb server rs1 192.168.1.1
ACOS(config-real server)# port 80 tcp
ACOS(config-real server-node port)# port 53 udp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
ACOS(config)# slb server rs2 192.168.1.2
ACOS(config-real server)# port 80 tcp
ACOS(config-real server-node port)# port 53 udp
ACOS(config-real server-node port)# exit
```

```
ACOS(config-real server)# exit
```

Then, configure the service group “SG_TCP” and add the servers (port 80 only) as members in the group. Because round robin load balancing is the default algorithm; the method command is not necessary.

```
ACOS(config)# slb service-group SG_TCP tcp
ACOS(config-slb svc group)# member rs1 80
ACOS(config-slb svc group-member:80)# exit
ACOS(config-slb svc group)# member rs2 80
ACOS(config-slb svc group-member:80)# exit
ACOS(config-slb svc group)# exit
```

Similarly, configure the service group “SG_UDP” to group the UDP servers. This service group also uses round robin load balancing.

```
ACOS(config)# slb service-group SG_UDP udp
ACOS(config-slb svc group)# member rs1 53
ACOS(config-slb svc group-member:53)# exit
ACOS(config-slb svc group)# member rs2 53
ACOS(config-slb svc group-member:53)# exit
ACOS(config-slb svc group)# exit
```

The following is an example of removing a service-group configuration:

```
ACOS(config)# no slb service-group
```

The deletion of the SLB objects is dependent on the global object deletion configuration. For more information, see *Disabling the Deletion of Referenced Objects* section in the [System Configuration and Administration Guide](#).

Wildcard VIPs, Ports, and Virtual Ports

A wildcard VIP is a VIP that does not have a specific IP address. Instead, wildcard VIPs have IP address 0.0.0.0 (for IPv4) or :: (for IPv6). Client requests sent to any IP address will be accepted when they are received at a wildcard VIP. Below is an example configuration for a wildcard VIP that will accept TCP requests on port 80:

```
ACOS(config)# slb virtual-server wildcard-vs1 0.0.0.0
ACOS(config-slb vserver)# port 80 tcp
ACOS(config-slb vserver-vport)# exit
ACOS(config-slb vserver)# exit
```

Wildcard VIPs allows you to configure a variable VIP depending on the ACL rules. To specify the subset of VIP addresses and ports for which a feature applies, you can use an Access Control List (ACL). ACLs also can specify the subset of clients allowed to access the VIPs, thus ensuring that only legitimate requests are allowed through to the real servers.

Wildcard VIPs can be used for any type of load balancing. Port 0 is used as a wildcard port to match any port number.

The maximum wildcard VIP limit is 200 for each partition. However, the number of wildcard VIPs per device is based on the system memory size. Also, the IPv4 and IPv6 VIPs are calculated separately. For example, with 64 GB of memory, you can configure up to 1200 IPv4 VIP and 1200 IPv6 VIP. For more information, see [Table 2](#).

Table 1 : Wildcard VIPs Limit on vThunder

Memory (GB RAM)	Maximum Wildcard VIP ACL
1 - 16	200
32	800
64	1200

Table 2 : Wildcard VIPs Limit on Hardware Platform

Memory (GB RAM)	Maximum Wildcard VIP
1 - 24	200
32	800
64 - 128	1200

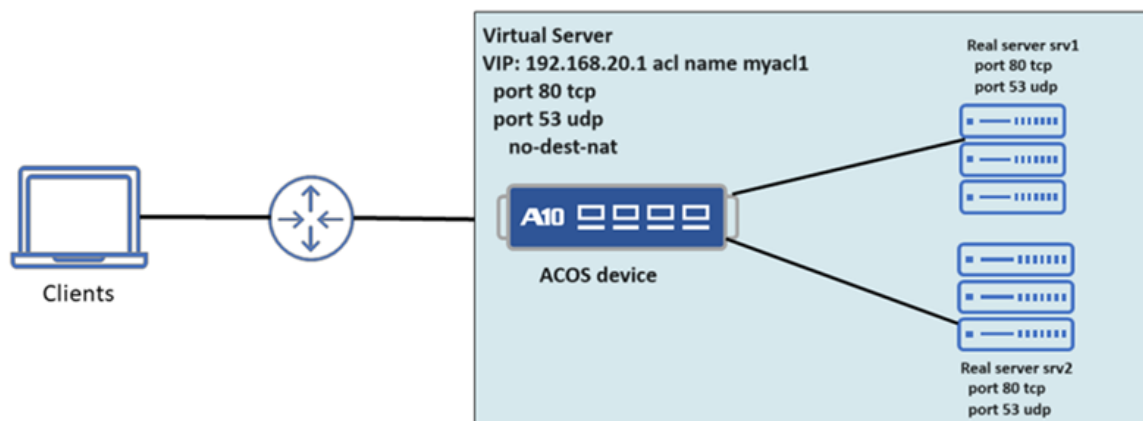
Normal VIP with ACL

A normal VIP with ACL functions like a wildcard VIP but has a specific IP address that is associated to an ACL. For example, 192.20.10.1 acl name myacl1. In this example, the VIP serves the same purpose as the destination IP in the ACL and is used for traffic matching. The source IP defined in the associated ACL is used to determine if

the traffic matches the defined virtual port and accordingly permits or denies the traffic.

You can configure multiple VIPs associated with ACLs in ACOS alongside wildcard VIPs associated with ACLs. The maximum configurable number depends on the platform and system memory size. The maximum configurable number of normal VIPs is greater than the maximum configurable number of wildcard VIPs. If traffic does not match any virtual port in a VIP with ACL, it uses the wildcard VIP for matching.

Figure 5 : VIP with ACL



For details on wildcard VIPs, see section [Wildcard VIP](#). For details on configuration of normal VIPs with ACL, see section [Transparent Proxy with Normal VIPs Associated to ACL](#).

Source Network Address Translation (NAT)

When load balancing traffic from the client to real servers, if the ACOS device changes the client's source IP address in the packets along with destination IP address translation (VIP to real server IP address), then it is referred to as source NAT. For traffic from servers to client, the ACOS device replaces the destination IP to be the actual client IP address.

Source NAT is required in cases such as:

- When the ACOS device is deployed in transparent mode in a multi-netted environment, for health checking of real servers and their application ports located in other subnets.

- When the ACOS device is deployed in either transparent one-arm mode or gateway one-arm mode.

Networking Modes

The ACOS device can be deployed on the following networking modes for server load balancing.

The following topics are covered:

Transparent (Switch) Mode	44
Gateway (Router) Mode	44
Virtual Wire Mode	45

Transparent (Switch) Mode

In transparent (switch) mode, the ACOS device behaves as a Layer 2 switch to provide support for switching functions such as MAC forwarding, ARP, Virtual LANs (VLANs), tagged and untagged interfaces, static and dynamic link aggregation (trunks), and multi-netted environments. In transparent mode, the ACOS device depends on a Layer 3 device to perform its routing.

In transparent mode, the ACOS device has a single IP interface. For multi-netted environments, you can configure multiple VLANs. IP Source NAT will be required, to allow health checking of servers and their application ports in other subnets.

[Transparent Mode Deployment](#) provides deployment examples.

Gateway (Router) Mode

Gateway (router) mode provides Layer 2 functionality as transparent mode and performs Layer 3 features, such as IP NAT, Access-lists, static routes, RIP, OSPF, IS-IS, PBR, and BGP. Layer 3 features provide flexibility when integrating the ACOS device into the network. ACOS devices in route mode supports separate IP addresses on each data interface. [Gateway Mode Deployment](#) describes gateway mode and provides deployment examples.

Virtual Wire Mode

ACOS supports virtual wire which consists of two interfaces that are logically bridged, forming a virtual wire pair. For packets that traverse the virtual wire endpoints, there is no Layer 2 modification, and the endpoints do not participate in MAC learning, ARP, or route lookups. The Thunder virtual wire deployment is an IP-Less configuration and virtual wire endpoints can be configured as an Ethernet, trunk, or virtual wire Ethernet group.

NOTE: Transparent and router mode are global to ACOS, whereas virtual wire mode is local and only applies to the virtual wire interfaces that are configured as endpoints. A set of interfaces can be used in virtual wire mode, while the remaining Ethernet ports can be configured for routed or transparent mode.

For detailed virtual wire configuration information, refer to the Networking Configuration Guide. For SLB related commands see [Real Server in Virtual Mode](#)

Deployment Modes

The ACOS device can be deployed in the following modes for both networking modes.

The following topics are covered:

Inline Mode	45
One-Arm Mode	46

Inline Mode

In-line topology is a physical topology where an ACOS device has at least two physical connections in the network; one data Ethernet port is connected to an “upstream” router or switch and another data Ethernet port is connected to a “downstream” switch or servers. In this topology, all traffic passes through the ACOS device, which can increase the load on the ACOS device.

One-Arm Mode

In one-arm mode, the ACOS device is connected to a switch or router like an extended arm. All real servers and applications configured for load balancing are assigned private IP addresses; all other servers and applications can be assigned public IP addresses with the default gateway pointing to the switch or router. Only traffic that is destined for the VIP will go to the ACOS device. Because not all traffic is passed through the ACOS device, one-arm topologies are often used for better performance or server throughput.

Part I: Application Delivery Controller Deployment

This part describes common application delivery and server load balancing deployment options.

The following topics are covered:

- [Transparent Mode Deployment](#)
- [Gateway Mode Deployment](#)
- [Direct Server Return \(DSR\) SLB Deployment](#)
- [Route Health Injection Deployment](#)
- [Network Address Translation](#)
- [Dynamic Real Server Creation Using DNS](#)
- [Virtual Port Ranges](#)
- [Mapping Virtual IP Addresses and Real Ports](#)

Transparent Mode Deployment

This section provides examples of Layer 2 transparent mode SLB deployment.

The following topics are covered:

Transparent Inline Mode	49
Transparent Mode in Multinetted Mode	54

Transparent Inline Mode

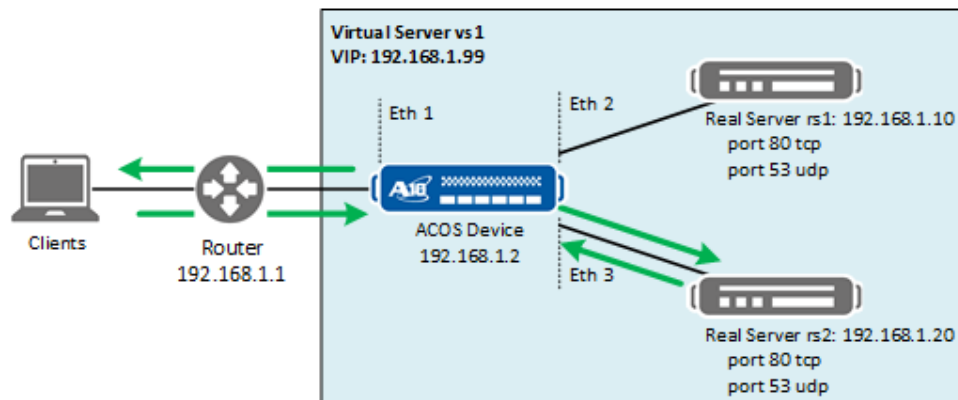
This topic provides an overview of transparent inline mode topology.

In this example, the ACOS device is inserted directly between the gateway router and the real servers. The ACOS device and real servers are all in the same subnet and all use the router as their default gateway.

The arrows illustrate the flow of traffic.

- The client requests for virtual server 192.168.1.99 are routed by the Layer 3 router to the ACOS device.
- ACOS device selects a real server and sends the request to that server.
- The server reply sends the response through the ACOS device back to the client.

Figure 6 : Transparent Inline Mode Topology



This example does not use Layer 3 Network Address Translation (NAT) but uses the default SLB NAT settings. For a description, see [Network Address Translation](#).

CLI Configuration

This topic describes the CLI configuration of transparent inline mode.

Configure the Default Route and Enable Interfaces

The following commands configure the default gateway and interfaces:

```
ACOS(config)# ip address 192.168.1.2 /24
ACOS(config)# ip default-gateway 192.168.1.1
ACOS(config)# interface ethernet 1
ACOS(config-if:ethernet:1)# enable
ACOS(config-if:ethernet:1)# interface ethernet 2
ACOS(config-if:ethernet:2)# enable
ACOS(config-if:ethernet:2)# interface ethernet 3
ACOS(config-if:ethernet:3)# enable
ACOS(config-if:ethernet:3)# exit
```

SLB Configuration – Real Servers

The following commands configure the real servers and ports (80 for TCP traffic, and 53 for UDP):

```
ACOS(config)# slb server rs1 192.168.1.10
ACOS(config-real server)# port 80 tcp
ACOS(config-real server-node port)# port 53 udp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
ACOS(config)# slb server rs2 192.168.1.20
ACOS(config-real server)# port 80 tcp
ACOS(config-real server-node port)# port 53 udp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
```

SLB Configuration – Service Groups

The following commands configure the service groups, one for TCP and a second for UDP. Round-robin will be the load balancing algorithm used for both service groups.

```
ACOS(config)# slb service-group SG_TCP tcp
ACOS(config-slb svc group)# method round-robin
ACOS(config-slb svc group)# member rs1 80
ACOS(config-slb svc group-member:80)# member rs2 80
ACOS(config-slb svc group-member:80)# exit
ACOS(config-slb svc group)# exit
```

```
ACOS(config)# slb service-group SG_UDP udp
ACOS(config-slb svc group)# method round-robin
ACOS(config-slb svc group)# member rs1 53
ACOS(config-slb svc group-member:80)# member rs2 53
ACOS(config-slb svc group-member:80)# exit
ACOS(config-slb svc group)# exit
```

SLB Configuration – Virtual Server

The following commands configure the virtual server and VIP, and add service groups to the VIP:

```
ACOS(config)# slb virtual-server vip1 192.168.1.99
ACOS(config-slb vserver)# port 80 tcp
ACOS(config-slb vserver-vport)# service-group SG_TCP
ACOS(config-slb vserver-vport)# exit
ACOS(config-slb vserver)# port 53 udp
ACOS(config-slb vserver-vport)# service-group SG_UDP
```

GUI Configuration

This topic describes the GUI configuration of transparent inline mode.

Interface Configuration

To configure the IP address and default gateway:

1. Navigate to **Network >> Interfaces >> Transparent**.
2. Enter **192.168.1.2** in the IP Address field
3. Enter **255.255.255.0** in the IP Mask field.
4. Enter **192.168.1.1** in the Default Gateway field.
5. Click **Configure**.

To enable the interfaces,

1. Navigate to **Network >> Interfaces > LAN**.
2. Click **Edit** in the Actions column for interface e1.
3. On the Update Ethernet page, select **Enable** in the Status field.

4. Click **Update**.
5. Repeat this procedure for the other interfaces (e2 and e3).

SLB Configuration - Real Servers

The following screen examples show the GUI pages for basic SLB configuration.

Configure the real servers:

1. Navigate to **ADC >> SLB >> Servers**.
2. Click **Create**.
3. Enter **rs1** in the Name field.
4. Enter **192.168.1.10** in the Host field.
5. In the Port section, click **Create**. In the Update Port page:
 - a. Enter **80** in the Port Number field.
 - b. Select **TCP** from the drop-down list in the Protocol field (should be the default value).
 - c. Click **Create**.
 - d. Repeat this procedure to create a second port, number 53 for UDP.
6. Click **Update**.
7. Repeat this procedure to create server rs2 with the IP address 192.168.1.20.

SLB Configuration – Service Group

Configure the service groups.

1. Navigate to **ADC >> SLB >> Service Groups**.
2. Click **Create**.
3. Enter **SG_TCP** in the Name field.
4. Verify that “TCP” is the protocol shown in the Protocol field (should be the default value).
5. Verify that “Round Robin” is shown in the Algorithm field (should be the default value).
6. Click **Create** in the Member pane. On the Create Member page:

7. Select server **rs1** from the drop-down list in the Server field.
8. Enter **80** in the port field.
9. Click **Create**.
10. Repeat to add port server rs2, port 80 as a member to the group.
11. Click **Update**.
12. Repeat this procedure to create a second service group called SG_UDP, with port 53 of both servers rs1 and rs2 as members of the group.

SLB Configuration - Virtual Server

Configure the virtual servers.

1. Navigate to **ADC >> SLB >> Virtual Servers**.
2. Click **Create**.
3. Enter **vip1** in the Name field.
4. Enter **192.168.1.99** in the IP Address field.
5. Click **Create** in the Virtual Port pane. On the next page:
6. Verify TCP is selected in the Protocol field.
7. Enter **80** in the Port field.
8. Select **SG_TCP** from the drop-down list in the Service Group field.
9. Click **Create**.
10. Repeat this procedure to add service group SG_UDP to the virtual server; be sure to select port 53 and UDP as the protocol.
11. Click **Update**.

View Services Map

View the services map to verify that the configuration is correct.

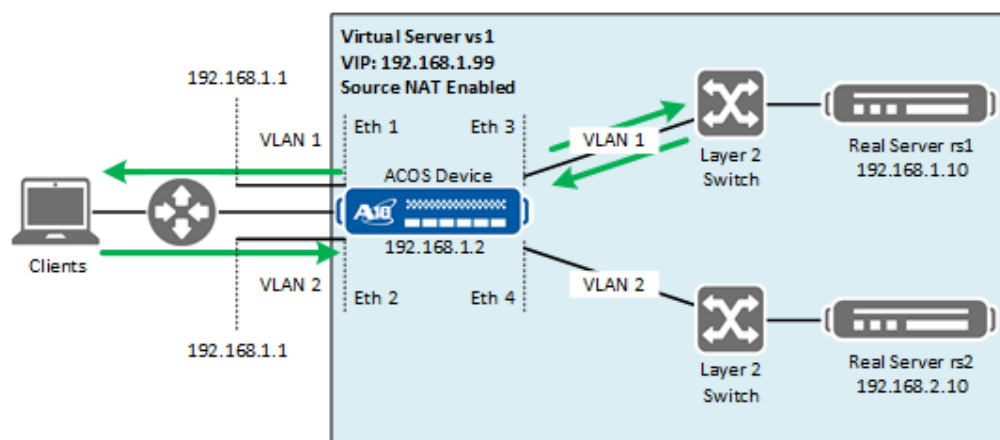
1. Navigate to **Dashboard >> Services Map**.
2. Select or search for **vip1** in the left-side column.

Transparent Mode in Multinetted Mode

This topic describes the overview of transparent mode in a multinetted topology.

[Figure 7](#) displays an ACOS device deployed in transparent mode in a multinetted environment.

Figure 7 : Transparent Mode in Multinetted Environment



Each server uses the router as its default gateway, but at a different subnet address. The arrows show the traffic flow for client-server traffic; in this example, between clients and server 192.168.1.10.

To enable the ACOS device to pass traffic for multiple subnets, the device is configured with multiple VLANs. The interfaces in subnet 192.168.1.x are in VLAN 1. The interfaces in the 192.168.2.x subnet are in VLAN 2.

NOTE: In this example, each ACOS interface is in only one VLAN and can therefore be untagged. the ACOS device could be connected to the router by a single link, in which case the ACOS link with the router would be in two VLANs and would need to be tagged in at least one of the VLANs. (If an interface is in multiple VLANs, the interface can be untagged in only one of the VLANs.)

Layer 3 IP Source NAT

The default SLB NAT settings allow client traffic to reach the server in the 192.168.2.x subnet, even though this is not the subnet that contains the ACOS device's IP address.

However, in a multinetted environment where the ACOS device is deployed in transparent mode, source NAT is required, to allow health checking of server 192.168.2.10 and its application port.

In this example, an address pool containing a range of addresses in the 192.168.2.x subnet is configured. The pool configuration includes the default gateway for the 192.168.2.x subnet (192.168.2.1). Without a gateway specified for the NAT pool, the ACOS device would attempt to send reply traffic using its own gateway (192.168.1.x), which is in a different subnet. The NAT configuration also includes enabling source NAT on the service port (in this example, 80) on the virtual server.

The ACOS device initiates health checks using the last (highest numbered) IP address in the pool as the source IP address. In addition, the ACOS device will only respond to control traffic (for example, management and ICMP traffic) from the NATted subnet if the control traffic is sent to the last IP address in the pool.

CLI Configuration

This section describes the CLI configuration example of transparent mode in a multinetted environment.

Configure the Network Settings

The following commands configure the global IP address and default gateway:

```
ACOS(config)# ip address 192.168.1.2 /24
ACOS(config)# ipdefault-gateway 192.168.1.1
```

Enable the Interfaces

The following commands enable the Ethernet interfaces used in the example:

```
ACOS(config)# interface ethernet 1
ACOS(config-if:ethernet:1)# enable
ACOS(config-if:ethernet:1)# exit
ACOS(config)# interface ethernet 2
```

```
ACOS(config-if:ethernet:2)# enable
ACOS(config-if:ethernet:2)# exit
ACOS(config)# interface ethernet 3
ACOS(config-if:ethernet:3)# enable
ACOS(config-if:ethernet:3)# exit
ACOS(config)# interface ethernet 4
ACOS(config-if:ethernet:4)# enable
ACOS(config-if:ethernet:4)# exit
```

Configure the VLANs

The following commands configure the VLANs. By default, all ACOS Ethernet data ports are in VLAN 1, so the only configuration required in this example is to create a second VLAN and add ports to it. The ports you add to other VLANs are automatically removed from VLAN 1.

```
ACOS(config)# vlan 2
ACOS(config-vlan:2)# untagged ethernet 2
ACOS(config-vlan:2)# untagged ethernet 4
ACOS(config-vlan:2)# exit
```

Configure the Source NAT Pool

The following command configures a pool of IP addresses for use by source NAT. The pool is in the same subnet as real server 192.168.2.10.

```
ACOS(config)# ip nat pool pool1 192.168.2.100 192.168.2.101 netmask /24
gateway 192.168.2.1
```

SLB Configuration – Real Servers

The following commands configure the real servers:

```
ACOS(config)# slb server rs1 192.168.1.10
ACOS(config-real server)# port 80 tcp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
ACOS(config)# slb server rs2 192.168.2.10
ACOS(config-real server)# port 80 tcp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
```

SLB Configuration – Service Group

The following commands configure the service group.

```
ACOS(config)# slb service-group sg-web tcp
ACOS(config-slb svc group)# member rs1 80
ACOS(config-slb svc group-member:80)# exit
ACOS(config-slb svc group)# member rs2 80
ACOS(config-slb svc group-member:80)# exit
ACOS(config-slb svc group)# exit
```

SLB Configuration – Virtual Server

The following commands configure the virtual server. The `source-nat` command enables the IP address pool configured above to be used for NATting health check traffic between the ACOS device and the real server.

```
ACOS(config)# slb virtual-server vip1 192.168.1.99
ACOS(config-slb vserver)# port 80 fast-http
ACOS(config-slb vserver-vport)# source-nat pool pool1
ACOS(config-slb vserver-vport)# service-group sg-web
ACOS(config-slb vserver-vport)# exit
ACOS(config-slb vserver)# exit
```

GUI Configuration

This section describes the GUI configuration of transparent mode in a multinetted.

Create the VLAN

To create the VLAN:

1. Navigate to **Network > VLAN**.
2. Click **Create**.
3. Enter **2** in the VLAN ID field.
4. In the Untagged Ethernet field, select 2 and 4.
5. Click **Create VLAN**.

Create the Source NAT Pool

To create the source NAT pool:

1. Navigate to **ADC >> IP Source NAT**.
2. Click **Create**.
3. Enter **pool1** in the Name field.
4. Enter **192.168.2.100** in the Start Address field.
5. Enter **192.168.2.101** in the End Address field.
6. Enter **255.255.255.0** in the Netmask field.
7. Click **Create**.

Interface Configuration

To configure the IP address and default gateway:

1. Navigate to **Network >> Interfaces >> Transparent**.
2. Enter **192.168.1.2** in the IP Address field
3. Enter **255.255.255.0** in the IP Mask field.
4. Enter **192.168.1.1** in the Default Gateway field.
5. Click **Configure**.

To enable the interfaces,

1. Navigate to **Network >> Interfaces > LAN**.
2. Click **Edit** in the Actions column for interface e1.
3. On the Update Ethernet page, select **Enable** in the Status field.
4. Click **Update**.

Repeat this procedure for the other interfaces (e2, e3, and e4).

SLB Configuration – Real Servers

The following screen examples show the GUI pages for basic SLB configuration.

Configure the real servers:

1. Navigate to **ADC >> SLB >> Servers**.
2. Click **Create**.
3. Enter **rs1** in the Name field.

4. Enter **192.168.1.10** in the Host field.
5. In the Port section, click **Create**. In the Update Port page:
 - a. Enter **80** in the Port Number field.
 - b. Select **TCP** from the drop-down list in the Protocol field (should be the default value).
 - c. Click **Create**.
6. Click **Update**.

Repeat this procedure to create server rs2 with the IP address 192.168.2.10.

SLB Configuration – Service Group

Configure the service groups.

1. Navigate to **ADC >> SLB >> Service Groups**.
2. Click **Create**.
3. Enter **sg-web** in the Name field.
4. Verify that “TCP” is the protocol shown in the Protocol field (should be the default value).
5. Click **Create** in the Member pane. On the Create Member page:
 - a. Select server **rs1** from the drop-down list in the Server field.
 - b. Enter **80** in the port field.
 - c. Click **Create**.
 - d. Repeat to add port server rs2, port 80 as a member to the group.
6. Click **Update**.

SLB Configuration – Virtual Server

Configure the virtual servers.

1. Navigate to **ADC >> SLB >> Virtual Servers**.
2. Click **Create**.
3. Enter **vip1** in the Name field.
4. Enter **192.168.1.99** in the IP Address field.

5. Click **Create** in the Virtual Port pane. On the next page:
 - a. Verify TCP is selected in the Protocol field.
 - b. Enter **80** in the Port field.
 - c. Select **sg-web** from the drop-down list in the Service Group field.
 - d. Click **Create**.
6. Click **Update**.

View Services Map

View the services map to verify that the configuration is correct.

1. Navigate to **Dashboard >> Services Map**.
2. Select or search for **vip1** in the left-side column.

Gateway Mode Deployment

This section provides examples of Layer 3 gateway mode SLB deployment.

The following topics are covered:

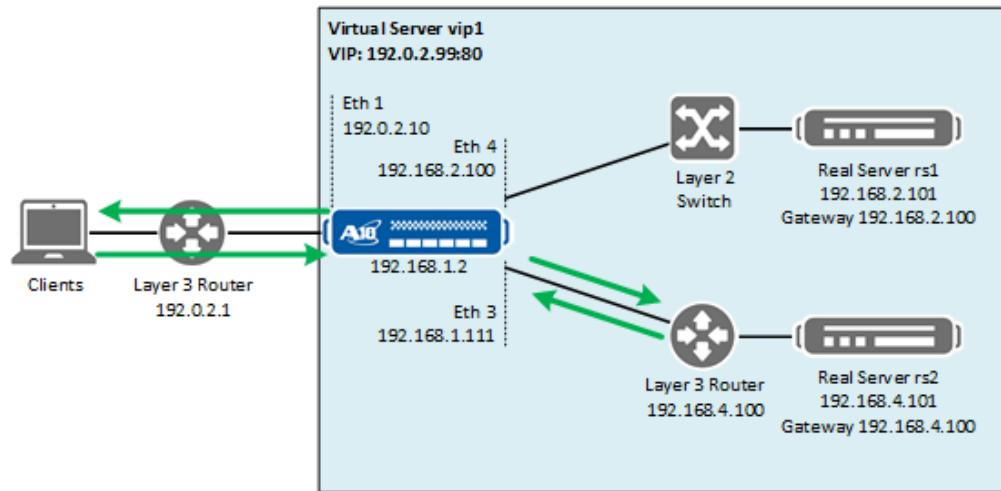
Gateway Inline Deployment	62
Gateway One-Arm Deployment	67

Gateway Inline Deployment

This topic describes the overview of gateway inline deployment topology.

[Figure 8](#) displays an ACOS device deployed in gateway mode.

Figure 8 : ACOS Deployment Example – Gateway Mode



The arrows display client-server traffic flow; in this example, between clients and server 192.168.4.101. Real servers can reach the database server through the ACOS device just as they would through any other router. Replies to clients still travel from the real servers through the ACOS device back to the client.

In this example, the ACOS device has separate IP interfaces in different subnets on each interface connected to the network. The ACOS device can be configured with static IP routes and enabled to run routing protocols such as OSPF and IS-IS. The example configures a static route as the default route through 192.0.2.1.

Although this example shows single physical links, you could use trunks as physical links. You also could use multiple VLANs. In this case, the IP addresses would be configured on Virtual Ethernet (VE) interfaces, one per VLAN, instead of being configured on individual Ethernet ports.

NOTE: For topologies where Thunder devices may be handling a large amount of L3 forwarding traffic, it is recommended to configure a Wildcard VIP or an interface level ACL (permit IP any any). For more information, refer [Wildcard VIP](#).

Since the ACOS device is a router in this deployment, downstream devices can use the ACOS device as their default gateway. The router connected to port 3 would use 192.168.1.111 as its default gateway, and the Layer 2 switch connected to 192.168.2.100 would use that address as its default gateway.

If a pair of ACOS devices in a VRRP-A configuration is used, the downstream devices would use a floating IP address shared by the two ACOS devices as their default gateway. (For more on VRRP-A, see the *Configuring VRRP-A High Availability* guide.)

Source NAT is not required for this configuration. The ACOS device can send health checks to the real servers and receive the replies without NAT.

CLI Configuration

This topic describes the CLI configuration example of gateway inline mode.

Interface Configuration

These commands enable the Ethernet interfaces used in the example and configure IP addresses on them:

```
ACOS(config)# interface ethernet 1
ACOS(config-if:ethernet:1)# enable
ACOS(config-if:ethernet:1)# ip address 192.0.2.10 /24
ACOS(config-if:ethernet:1)# exit
ACOS(config)# interface ethernet 3
ACOS(config-if:ethernet:3)# enable
ACOS(config-if:ethernet:3)# ip address 192.168.1.111 /24
ACOS(config-if:ethernet:3)# exit
ACOS(config)# interface ethernet 4
ACOS(config-if:ethernet:4)# enable
ACOS(config-if:ethernet:4)# ip address 192.168.2.100 /24
ACOS(config-if:ethernet:4)# exit
```

Default Route Configuration

The following command configures the default route through 192.0.2.1:

```
ACOS(config)# ip route 0.0.0.0 /0 192.0.2.1
```

SLB Configuration - Real Servers

```
ACOS(config)# slb server rs1 192.168.2.101
ACOS(config-real server)# port 80 tcp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
ACOS(config)# slb server rs2 192.168.4.101
ACOS(config-real server)# port 80 tcp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
```

SLB Configuration - Service Group

```
ACOS(config)# slb service-group sg-web tcp
ACOS(config-slb svc group)# member rs1 80
ACOS(config-slb svc group-member:80)# exit
ACOS(config-slb svc group)# member rs2 80
ACOS(config-slb svc group-member:80)# exit
ACOS(config-slb svc group)# exit
```

SLB Configuration - Virtual Server

```
ACOS(config)# slb virtual-server vip1 192.0.2.99
ACOS(config-slb vserver)# port 80 http
ACOS(config-slb vserver-vport)# service-group sg-web
ACOS(config-slb vserver-vport)# exit
ACOS(config-slb vserver)# exit
```

GUI Configuration

This topic describes the GUI configuration example of gateway inline mode.

Interface Configuration

To configure the interfaces:

1. Navigate to **Network >> Interfaces >> LAN**.
2. Click **Edit** in the Actions column for interface e1.

3. Click **Enable** in the Status field.
4. Expand the IP pane.
5. Enter **192.0.2.10** in the “IPv4 Address” column of the table in the IP Address field.
6. Enter **255.255.255.0** in the “Netmask” column of the table in the IP Address field.
7. Click **Update**. (not shown in the figure).
8. Repeat the procedure to configure the other interfaces and IP addresses.

Default Route Configuration

To configure the static route:

1. Navigate to **Network >> Routes >> Static IPv4 Routes**.
2. Click **Create**.
3. Enter **0.0.0.0** in the IP Dest Address field.
4. Enter **/0** in the IP Mask field.
5. Enter **192.0.2.1** in the “Next Hop IP” column of the IP Next Hop field, and click **Add**.
6. Click **Create Route**.

SLB Configuration - Real Servers

Configure the real servers:

1. Navigate to **ADC >> SLB >> Servers**.
2. Click **Create**.
3. Enter **rs1** in the Name field.
4. Enter **192.168.2.101** in the Host field.
5. In the Port section, click **Create**. In the Update Port page:
 - a. Enter **80** in the Port Number field.
 - b. Select **TCP** from the drop-down list in the Protocol field (should be the default value).
 - c. Click **Create**.

6. Click **Update**.
7. Repeat this procedure to create server rs2 with the IP address 192.168.4.101.

SLB Configuration - Service Group

Configure the service groups.

1. Navigate to **ADC >> SLB >> Service Groups**.
2. Click **Create**.
3. Enter **sg-web** in the Name field.
4. Verify that TCP is the protocol shown in the Protocol field (should be the default value).
5. Click **Create** in the Member pane. On the Create Member page:
 - a. Select server **rs1** from the drop-down list in the Server field.
 - b. Enter **80** in the port field.
 - c. Click **Create**.
 - d. Repeat to add server rs2 as a member to the group.
6. Click **Update**.

SLB Configuration - Virtual Server

Configure the virtual servers.

1. Navigate to **ADC >> SLB >> Virtual Servers**.
2. Click **Create**.
3. Enter **vip1** in the Name field.
4. Enter **192.0.2.99** in the IP Address field.
5. Click **Create** in the Virtual Port pane. On the next page:
 - a. Verify TCP is selected in the Protocol field.
 - b. Enter **80** in the Port field.
 - c. Select **sg-web** from the drop-down list in the Service Group field.
 - d. Click **Create**.
6. Click **Update**.

View Services Map

View the services map to verify that the configuration is correct.

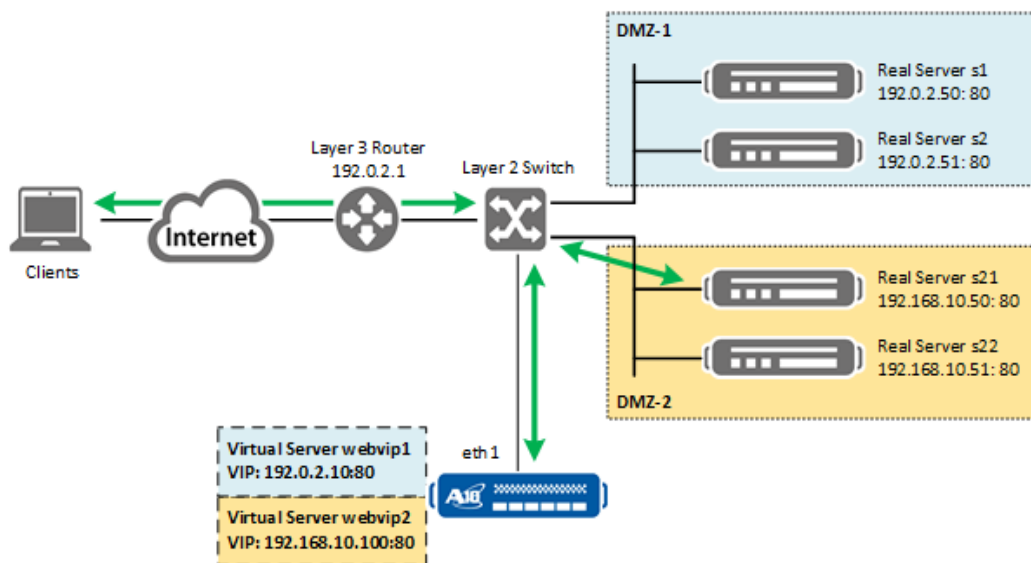
1. Navigate to **Dashboard >> Services Map**.
2. Select or search for **vip1** in the left-side column.

Gateway One-Arm Deployment

One-arm deployment allows the ACOS device to be added to the network without inserting the device directly into the traffic path between clients and servers. [Figure 9](#) shows an example of an ACOS device deployed in route mode in a one-arm topology.

This example includes use of Access Control Lists (ACLs) to add a layer of security on top of the basic deployment. An alternative is to use L3V partitions, which allow the ACOS device to be partitioned into multiple logical devices with their own independent network resources. For more information, see the *Configuring Application Delivery Partitions* Guide.

Figure 9 : ACOS Deployment Example – Route Mode One-Arm Deployment



Traffic Flow

The arrows display client-server traffic flow between clients and server 192.168.10.51.

1. A client sends a request to 192.168.10.100:80.
2. The Layer 3 router forwards the request to the ACOS device.
3. The ACOS device selects a server within the VIP's service group, and forwards the request to the server.
4. The server reply is routed back to the ACOS device, which then forwards the reply to the client.

ACOS Interface Configuration

The ACOS device has a single physical connection to the network, through the Layer 2 switch. Two VIPs are configured on the ACOS device:

- 192.0.2.10:80 – The ACOS device load balances requests to this VIP to the servers in DMZ1, on subnet 192.0.2.0 /24.
- 192.168.10.100:80 – The ACOS device load balances requests to this VIP to the servers in DMZ2, on subnet 192.168.10.0 /24.

The Layer 3 router routes requests to either VIP to the ACOS device. The ACOS physical interface (Ethernet port 1) connected to the Layer 2 switch is configured with a separate IP interface to each real server subnet:

- 192.0.2.2 – This IP interface is configured on Virtual Ethernet (VE) interface 10, on VLAN 10.
- 192.168.10.2 – This IP interface is configured on VE interface 20, on VLAN 20.

A default route on the ACOS device routes server reply traffic through the Layer 3 router to clients.

Traffic Blocking Between VLANs

Ethernet port 1 is an 802.1Q-tagged member of each VLAN. As standard behavior, Layer 2 traffic is not forwarded between the VLANs.

To also prevent Layer 3 traffic from being forwarded between the VLANs, an Access Control List (ACL) is used. The ACL has the following rules:

- Deny IP traffic from 192.0.2.x to 192.168.10.x
- Deny IP traffic from 192.168.10.x to 192.0.2.x

The ACL is applied to each VLAN's VE.

Network Address Translation

The ACOS device uses source NAT to communicate with the servers. A separate pool is configured for each server subnet. Source NAT ensures that server replies pass back through the ACOS device before being forwarded to clients.

CLI Configuration

This topic describes the CLI configuration of the gateway one-arm deployment mode.

ACL Configuration

The following commands configure ACL 101:

```
ACOS(config)# access-list 101 deny ip 192.0.2.10 0.0.0.255  
192.168.10.100 0.0.0.255 log  
ACOS(config)# access-list 102 deny ip 192.168.10.100 0.0.0.255  
192.0.2.10 0.0.0.255 log
```

The ACL blocks Layer 3 traffic between VLANs. The `log` option enables generation of log messages when traffic matches the ACL and is dropped.

The `log` option enables logging for this ACL, but logging still must be enabled on the system level. Logging is disabled by default. To configure logging, see the “Basic Setup” section in the System Configuration and Administration Guide.

Network Configuration

The following commands enable Ethernet interface 1:

```
ACOS(config)# interface ethernet 1  
ACOS(config-if:ethernet:1)# enable  
ACOS(config-if:ethernet:1)# exit
```

The following commands configure VLANs 10 and 20:

```
ACOS(config)# vlan 10  
ACOS(config-vlan:10)# tagged ethernet 1  
ACOS(config-vlan:10)# router-interface ve 10  
ACOS(config-vlan:10)# exit  
ACOS(config)# vlan 20  
ACOS(config-vlan:20)# tagged ethernet 1  
ACOS(config-vlan:20)# router-interface ve 20  
ACOS(config-vlan:20)# exit
```

These commands configure IP interfaces on the VLAN's VEs. The **access-list** commands bind ACL 101 to the inbound traffic direction on the interfaces. The ACL does not take effect until it is bound to the interfaces.

```
ACOS(config)# interface ve 10  
ACOS(config-if:ve:10)# ip address 192.0.2.2 /24  
ACOS(config-if:ve:10)# access-list 101 in  
ACOS(config-if:ve:10)# exit  
ACOS(config)# interface ve 20  
ACOS(config-if:ve:20)# ip address 192.168.10.2 /24  
ACOS(config-if:ve:20)# access-list 102 in  
ACOS(config-if:ve:20)# exit
```

The following command configures the default route:

```
ACOS(config)# ip route 0.0.0.0 /0 192.0.2.1
```

SLB Configuration

The following commands configure the source NAT pools:

```
ACOS(config)# ip nat pool dmz1 192.0.2.200 192.0.2.200 netmask /24  
ACOS(config)# ip nat pool dmz2 192.168.10.200 192.168.10.200 netmask /24
```

The following commands configure the real servers:

```
ACOS(config)# slb server s1 192.0.2.50  
ACOS(config-real server)# port 80 tcp  
ACOS(config-real server-node port)# exit  
ACOS(config-real server)# exit  
ACOS(config)# slb server s2 192.0.2.51  
ACOS(config-real server)# port 80 tcp  
ACOS(config-real server-node port)# exit
```

```
ACOS(config-real server)# exit
ACOS(config)# slb server s21 192.168.10.50
ACOS(config-real server)# port 80 tcp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
ACOS(config)# slb server s22 192.168.10.51
ACOS(config-real server)# port 80 tcp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
```

The following commands configure the service groups:

```
ACOS(config)# slb service-group wbggroup1 tcp
ACOS(config-slb svc group)# member s1 80
ACOS(config-slb svc group-member:80)# exit
ACOS(config-slb svc group)# member s2 80
ACOS(config-slb svc group-member:80)# exit
ACOS(config-slb svc group)# exit
ACOS(config)# slb service-group wbggroup2 tcp
ACOS(config-slb svc group)# member s21 80
ACOS(config-slb svc group-member:80)# exit
ACOS(config-slb svc group)# member s22 80
ACOS(config-slb svc group-member:80)# exit
ACOS(config-slb svc group)# exit
```

The following commands configure the virtual servers:

```
ACOS(config)# slb virtual-server webvip1 192.0.2.10
ACOS(config-slb vserver)# port 80 http
ACOS(config-slb vserver-vport)# source-nat pool dmz1
ACOS(config-slb vserver-vport)# service-group wbggroup1
ACOS(config-slb vserver-vport)# exit
ACOS(config-slb vserver)# exit
ACOS(config)# slb virtual-server webvip2 192.168.10.100
ACOS(config-slb vserver)# port 80 http
ACOS(config-slb vserver-vport)# source-nat pool dmz2
ACOS(config-slb vserver-vport)# service-group wbggroup2
```

GUI Configuration

This topic describes GUI configuration of gateway one-arm deployment mode.

ACL Configuration

The ACL blocks Layer 3 traffic between VLANs. The Log option enables generation of log messages when traffic matches the ACL and is dropped.

The log option enables logging for this ACL, but logging still must be enabled on the system level. Logging is disabled by default. To configure logging, see the “Basic Setup” section in the *System Configuration and Administration Guide*.

This procedure configures an ACL to block traffic from 192.0.2.x (source IP address) to 192.168.10.x (destination IP address):

1. Navigate to **Security >> Access List >> Extended**.
2. Click **Create**.
3. Enter **101** in the ID field.
4. Click the **Entry** radio button in the Remark field.
5. Select **IP** in the Service field.
6. Select **Source Address** and **Subnet** in the Source Address field.
 - a. Enter **192.0.2.0** in the Address field.
 - b. Enter **0.0.0.255** in the Mask field.
7. Select **Destination Address** and **Subnet** in the Destination Address field.
 - a. Enter **192.168.10.0** in the Address field.
 - b. Enter **0.0.0.255** in the Mask field.
8. Select the checkbox in the Log field.
9. Click **Create**.
10. Repeat the procedure to configure a second ACL that will deny traffic from 192.168.10.x (source IP address) to 192.0.2.x (destination IP address).

Interface and VLAN Configuration

To enable the physical interface:

1. Navigate to **Network >> Interfaces >> LAN**.
2. Click **Edit** in the Actions columns for interface e1.

3. In the Status field, click **Enable**.
4. Click **Update**.

To create the VLAN:

1. Navigate to **Network >> VLAN**.
2. Click **Create**.
3. Enter **10** in the VLAN ID field.
4. Click the checkbox in the Create Virtual Interface field.
5. Select 1 from the list of interfaces in the Tagged Ethernet field.
6. Click **Create VLAN**.
7. Repeat the procedure to create VLAN 20.

Default Route Configuration

The following GUI page configures the default route.

1. Navigate to **Network >> Routes >> Static IPv4 Routes**.
2. Click **Create**.
3. Enter **0.0.0.0** in the IP Dest Address field.
4. Enter **/0** in the IP Mask field.
5. Enter **192.0.2.1** in the “Next Hop IP” column of the IP Next Hop field, and click **Add**.
6. Click **Create Route**.

Source NAT Pool Configuration

The following GUI pages configure the IP source NAT pools.

1. Navigate to **ADC >> IP Source NAT >> IPv4 Pools**.
2. Click **Create**.
3. Enter **dmz1** in the Name field.
4. Enter **192.0.2.200** in the Start Address field.
5. Enter **192.0.2.200** in the End Address field.
6. Enter **255.255.255.0** in the Netmask field.

7. Click **Create**.
8. Repeat the procedure to create the pool for DMZ2 servers.

SLB Configuration – Real Servers

The following GUI pages configure the real servers.

1. Navigate to **ADC >> SLB >> Servers**.
2. Click **Create**.
3. Enter **s1** in the Name field.
4. Enter **192.0.2.50** in the Host field.
5. In the Port section, click **Create**. In the Update Port page:
 - a. Enter **80** in the Port Number field.
 - b. Select **TCP** from the drop-down list in the Protocol field (should be the default value).
 - c. Click **Create**.
6. Click **Update**.
7. Repeat this procedure to create the other servers and ports.

SLB Configuration – Service Groups

The following GUI pages configure the service groups.

1. Navigate to **ADC >> SLB >> Service Groups**.
2. Click **Create**.
3. Enter **wbgroup1** in the Name field.
 - a. Verify that TCP is the protocol shown in the Protocol field (should be the default value).
 - b. Click **Create** in the Member pane. On the Create Member page:
 - c. Select server **s1** from the drop-down list in the Server field.
 - d. Enter **80** in the port field.
 - e. Click **Create**.
4. Repeat to add server s2 as a member to the group.

5. Click **Update**.
6. Repeat this procedure to create service group **wbgroup2** with servers **s21** and **s22** as members.

SLB Configuration – Virtual Servers

The following GUI pages configure the virtual servers.

1. Navigate to **ADC >> SLB >> Virtual Servers**.
2. Click **Create**.
3. Enter **webvip1** in the Name field.
4. Enter **192.0.2.10** in the IP Address field.
5. Click **Create** in the Virtual Port pane. On the next page:
 - a. Verify TCP is selected in the Protocol field.
 - b. Enter **80** in the Port field.
 - c. Select **wbgroup1** from the drop-down list in the Service Group field.
 - d. Expand the General Fields section, then select **dmz1** from the drop-down list in the Source NAT Pool field.
 - e. Click **Create**.
6. Click **Update**.
7. Repeat the procedure to create **webvip2**, with 192.168.10.100 as the IP address, **wbgroup2** as the service group and **dmz2** as the source NAT pool.

View Services Map

View the services map to verify that the configuration is correct.

1. Navigate to **Dashboard >> Services Map**.
2. Select or search for **webvip1** and **webvip2** and select both in the left-side column.

Direct Server Return (DSR) SLB Deployment

This section provides additional deployment examples for Server Load Balancing (SLB).

The following topics are covered:

Direct Server Return in Transparent Mode	77
Direct Server Return in Route Mode	82
Direct Server Return in Mixed Layer 2/Layer 3 Mode	84
Layer 3 Direct Server Return (IP Tunneling)	88

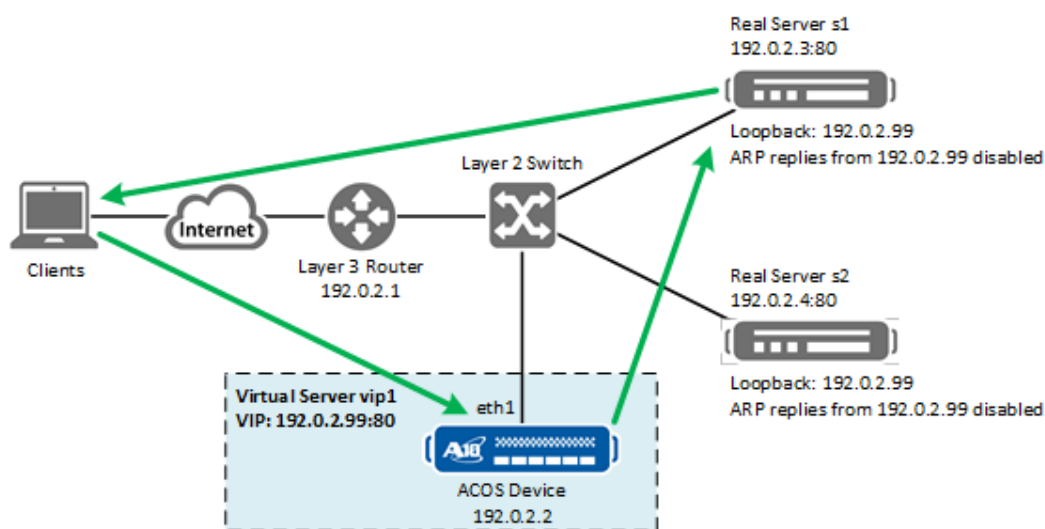
Direct Server Return in Transparent Mode

[Figure 10](#) displays an ACOS device deployed in a Direct Server Return (DSR) transparent mode.

In this example, an ACOS device attaches to the network in a “one-arm” configuration. A single link connects the device to the network. The link can be on a single Ethernet port or a trunk.

Arrows indicate client-server traffic flow. Client request traffic for virtual server IP address 192.0.2.99 is routed to the ACOS device. Server reply traffic does not pass back through the ACOS device.

Figure 10 : ACOS Deployment Example – DSR in Transparent Mode



The following topics are covered:

Requirements for DSR in Transparent Mode Deployment	78
DSR Health Checking	78
CLI Configuration	79
GUI Configuration	80

Requirements for DSR in Transparent Mode Deployment

This configuration has certain requirements:

- Requirements on the ACOS device:
 - The ACOS device, virtual server, and the real servers all must be in the same subnet.
 - The virtual server IP address must be configured as a loopback interface on each real server. (This is performed on the real server itself, not as part of the real server's configuration on the ACOS device.)
 - DSR must be enabled on the virtual service ports. (Enabling DSR is equivalent to disabling destination NAT on the return traffic. This allows real servers to respond directly to clients, bypassing the ACOS device and retaining the IP address of the real server in the response to the client.)
 - For IPv4 VIPs, DSR is supported on virtual port types (service types) TCP, UDP, and FTP. For IPv6 VIPs, DSR is supported on virtual port types TCP, and UDP.
- Requirements on the real server:
 - A loopback interface must be configured with the virtual server IP address.
 - ARP replies from the loopback interfaces must be disabled. (This applies to the loopback interfaces that have the virtual server IP address.)

DSR Health Checking

Layer 3 and Layer 4-7 health checks are supported in DSR configurations.

The target of the Layer 3 health checks can be the real IP addresses of the servers, or the virtual IP address.

- To send Layer 3 health checks to real server IP addresses, use the default Layer 3 health method (ICMP).
- To send Layer 3 health checks to a virtual IP address:
 - Configure an ICMP health method with the transparent option enabled, and with the alias address set to the virtual IP address.
 - Globally enable DSR health checking.

Layer 4-7 health checks are sent to the same IP address as Layer 3 health checks, then addressed to the specific protocol port. You can use the default TCP and UDP health monitors or configure new health monitors. The example uses the default TCP health monitor.

NOTE: In DSR route mode, we support health monitor configuration at the service-group configuration level. If you configure the health monitor at the real server, real server port, or service group member configuration level, it may not work in some network topology. For example, when there is one or more routers between ACOS and real server, the traffic may not send to the real server.

CLI Configuration

This topic describes the CLI configuration example of DSR in transparent mode.

Configuring the IP Address and Default Gateway

The following commands configure the global IP address and default gateway:

```
ACOS(config)# ip address 192.0.2.2 /24
ACOS(config)# ip default-gateway 192.0.2.1
```

Enabling the Ethernet Interfaces

The following commands enable the Ethernet interface connected to the clients and server:

```
ACOS(config)# interface ethernet 1
ACOS(config-if:ethernet:1)# enable
ACOS(config-if:ethernet:1)# exit
```

SLB Configuration - Real Servers

To configure real servers rs1 and rs2:

```
ACOS(config)# slb server rs1 192.0.2.3
ACOS(config-real server)# port 80 tcp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
ACOS(config)# slb server rs2 192.0.2.4
```

```
ACOS(config-real server)# port 80 tcp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
```

SLB Configuration - Service Group

To configure the service group sg-web with port 80 on rs1 and rs2 as members:

```
ACOS(config)# slb service-group sg-web tcp
ACOS(config-slb svc group)# member rs1 80
ACOS(config-slb svc group-member:80)# exit
ACOS(config-slb svc group)# member rs2 80
ACOS(config-slb svc group-member:80)# exit
ACOS(config-slb svc group)# exit
```

SLB Configuration - Virtual Server and Enabling DSR

To configure the virtual server vip1 and enable DSR:

```
ACOS(config)# slb virtual-server vip1 192.0.2.99
ACOS(config-slb vserver)# port 80 tcp
ACOS(config-slb vserver-vport)# service-group sg-web
ACOS(config-slb vserver-vport)# no-dest-nat
```

Configuration on the Real Servers

For DSR to work, a loopback interface with the IP address of the virtual server must be configured on each real server, and ARP replies from the loopback address must be disabled.

For example, on a Unix/Linux server:

```
ifconfig lo:0 192.0.2.99 netmask 255.255.255.255 -arp up
echo 1 > /proc/sys/net/ipv4/conf/all/arp_ignore
echo 1 > /proc/sys/net/ipv4/conf/eth0/arp_ignore
echo 2 > /proc/sys/net/ipv4/conf/all/arp_announce
echo 2 > /proc/sys/net/ipv4/conf/eth0/arp_announce
```

GUI Configuration

This topic describes the GUI configuration example of DSR in transparent mode.

Configure the IP Address and Default Gateway

To configure the IP address and default gateway:

1. Navigate to **Network >> Interfaces >> Transparent**.
2. Enter **192.0.2.2** in the IP Address field.
3. Enter **255.255.255.0** in the IP Mask field.
4. Enter **192.0.2.1** in the Default Gateway field.
5. Click **Configure**.

Enable Ethernet Interfaces

To enable the interfaces:

1. Navigate to **Network >> Interfaces >> LAN**.
2. On the menu bar, select the LAN tab, if not already displayed.
3. Select the checkbox in the row for e1, then click **Enable**.

SLB Configuration - Real Servers

To configure the real servers:

1. Navigate to **ADC >> SLB >> Servers**.
2. Click **Create**.
3. Enter **rs1** in the Name field.
4. Enter **192.0.2.3** in the Host field
5. In the Port section, click **Create**.
 - a. Enter **80** in the Port Number field.
 - b. Click **Create**.
 - c. Click **Update**.
6. Repeat this procedure to create server rs2, with the IP address 192.0.2.4.

SLB Configuration - Service Group

To configure the service group:

1. Navigate to **ADC >> SLB >> Service Groups**.
2. Click **Create**.

3. Enter **sg-web** in the Name field.
4. In the Member section, click **Create**.
5. Select **rs1** from the drop-down list in the Server field.
 - a. Enter **80** in the Port field.
 - b. Click **Create**.
6. Repeat to add server rs2 as a member.
7. Click **Update**.

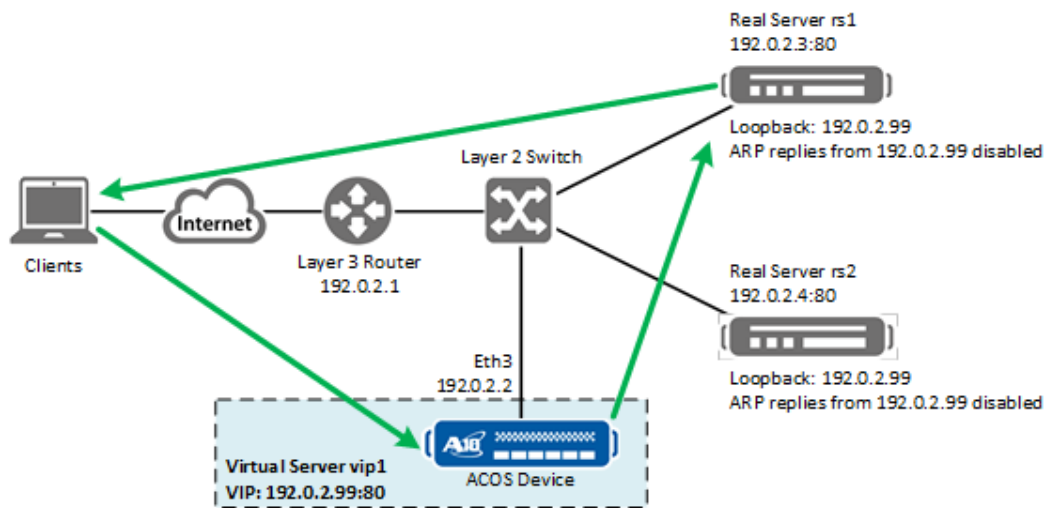
SLB Configuration - Virtual Port and Enabling DSR

1. Navigate to **ADC >> SLB >> Virtual Servers**.
2. Click **Create**.
3. Enter **vip1** in the Name field.
4. Enter **192.0.2.99** in the IP Address field.
5. In the Virtual Port section, click **Create**.
 - a. Enter **80** in the Port field.
 - b. Expand the General section, then click the checkbox in the No Dest NAT field.
 - c. Click **Create**.
6. Click **Update**.

Direct Server Return in Route Mode

[Figure 11](#) shows an example of an ACOS device deployed in a DSR configuration in route mode.

Figure 11 : ACOS Deployment Example – DSR in Route Mode



The configuration is similar to the DSR in transparent mode, except the ACOS device uses an IP interface configured on an individual Ethernet port instead of a global IP address.

The requirements for the ACOS device and real servers are the same as those for DSR in transparent mode. ([Direct Server Return in Transparent Mode](#))

CLI Configuration

This topic described the CLI configuration example of DSR in transparent mode.

The following configuration procedures differ from the instructions provided in [GUI Configuration](#). These commands enable the Ethernet interface used in the example and configure an IP address on it:

```
ACOS(config)# interface ethernet 3
ACOS(config-if:ethernet:3)# enable
ACOS(config-if:ethernet:3)# ip address 192.0.2.2 /24
ACOS(config-if:ethernet:3)# exit
```

The following command configures the default route through 192.0.2.1:

```
ACOS(config)# ip route 0.0.0.0 /0 192.0.2.1
```

GUI Configuration

This topic describes the GUI configuration example of DSR in transparent mode.

Configure an IP Address on the Ethernet Port

To configure an IP Address in the Ethernet interface:

1. Navigate to **Network > Interfaces >> LAN**.
2. In the Actions column, click the **Edit** link for interface e3.
3. Select the **Enabled** radio button in the Status field.
4. Expand the IP section.
 - a. Enter **192.0.2.2** in the IPv4 Address column of the IP Address field.
 - b. Enter **255.255.255.0** in the Netmask column of the IP Address field.
 - c. Click **Add**.
5. Click **Update**.

Configure a Default Route

To configure a default route:

1. Navigate to **Network >> Interfaces >> Transparent**.
2. Enter **0.0.0.0** in the IP Address and IP Mask fields.
3. Enter **192.0.2.1** in the Default Gateway field.
4. Click **Configure**.

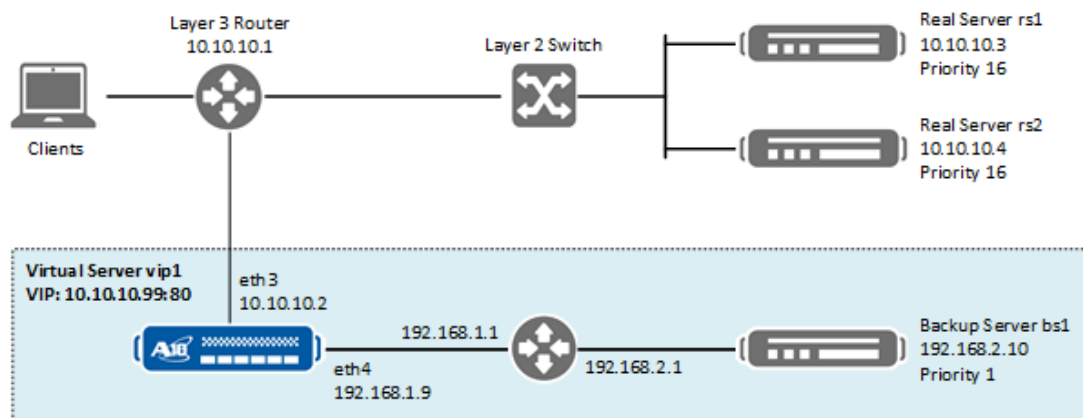
Direct Server Return in Mixed Layer 2/Layer 3 Mode

The ACOS device can use servers as backups in a DSR deployment. Backup servers are not required to be connected to the ACOS device at Layer 2 or in the same IP subnet. [Figure 12](#) displays a topology that includes a backup server in a different subnet. This backup server is used only when primary servers are unavailable.

In this example, two real servers are primary servers for VIP 10.10.10.99:80. They are in the same IP subnet as the ACOS device. Each of them is configured for DSR: destination NAT is disabled on the virtual port.

Another server, 192.168.2.10, is configured as a backup, to be used only if both primary servers are unavailable. Since the backup server is a valuable network resource, serving as a server farm backup is only one of its functions. It also used by other applications elsewhere in the network. The ACOS device can be configured to use the server as a backup to a DSR server farm, without changing the network topology.

Figure 12 : Backup Server in DSR Configuration



To deploy the backup server:

- In the service group, assign a higher priority to members for the primary servers, so that the member for the backup server has lower priority. By default, the ACOS device does not use the lower-priority (backup) server unless all primary servers are down. Use the same priority for all primary servers.
- Enable destination NAT on the backup server. By default, destination NAT is unset on real ports and set by the virtual port. Normally, destination NAT is disabled on virtual ports used for DSR. However, destination NAT needs to be enabled on the real port on the backup server.

Enabling destination NAT for the backup server allows the server to remain on a different subnet from the ACOS device and used for the VIP normally served by DSR. The backup server does not need to be moved to a Layer 2 connection to the ACOS device and the server's IP address does not need to be changed. It can remain on a different subnet from the ACOS device and the primary servers.

Destination NAT can not be set directly on an individual real port. To enable destination NAT on a real port, create a real port template and enable destination NAT in the template. You can bind the template to the real port itself, or to the service group member for the port.

- Templates bound directly to a port are applied to the port in all service groups using the port.
- When binding the template to a service group member, the template applies to ports only in the service group. The template does not apply to the same port when used in other service groups.

CLI Configuration

This topic described the CLI configuration example of DSR in Mixed L2 or L3 environment.

The following commands configure a server port template for the backup server and add members to the service group. Configuration commands for the component members are not included.

```
ACOS(config)# slb template port dsrbackup
ACOS(config-rport)# dest-nat
ACOS(config-rport)# exit
ACOS(config)# slb service-group sg-dsr tcp
ACOS(config-slb svc group)# member primarys1 80
ACOS(config-slb svc group-member:80)# priority 16
ACOS(config-slb svc group-member:80)# exit
ACOS(config-slb svc group)# member primarys2 80
ACOS(config-slb svc group-member:80)# priority 16
ACOS(config-slb svc group-member:80)# exit
ACOS(config-slb svc group)# member secondarys1 80
ACOS(config-slb svc group-member:80)# template dsrbackup
ACOS(config-slb svc group-member:80)# exit
ACOS(config-slb svc group)# exit
```

GUI Configuration

This topic describes the GUI configuration of DSR in Mixed L2 or L3 environment.

Configure a Port Template to Enable Destination NAT on the Backup Server's Port

1. Navigate to **ADC >> Templates >> SLB**.
2. Click **Create** and select **Port** from the drop-down list.
3. Enter a name for the template in the Name field.
4. To enable destination NAT, click the checkbox in the Destination NAT field.
5. Click **Create**.

Configure the Service Group

1. Navigate to **ADC >> SLB >> Service Groups**.
2. Click **Create** to add a new service group.
3. Enter the service group name.
4. Add the primary servers to the service group:
 - a. In the Member area of the window, click **Create**. The Create Member window appears.
 - b. Select a primary server from the Server drop-down list.
 - c. Enter the protocol port number in the Port field.
 - d. Enter **16** in the Priority field.
 - e. Click **Create**.
 - f. Repeat for the other primary server.
5. Add the backup server to the service group:
 - a. In the Member area of the service group window, click **Create**.
 - b. Select the backup server from the Server drop-down list.
 - c. Enter the protocol port number in the Port field.
 - d. Enter **1** in the Priority field.
 - e. Select the port template for the backup server from the Server Port Template drop-down list. This is the template configured earlier.
 - f. Click **Create**.
6. Click **Update**.

To set the priority values of the primary servers to a higher value than the backup server, re-add the members for the primary servers' ports, and use the `priority` option. Set the priority to a value higher than 1 (the default). Use the same priority value on each of the primary server's member ports.

To enable destination NAT on a service port within a service group, use the `dest-nat` option in a server port template, then bind that template to the server port in the service group.

Layer 3 Direct Server Return (IP Tunneling)

When DSR is enabled, the ACOS device sends client requests to back-end servers in an IP-in-IP tunnel. These servers respond directly to clients, bypassing the ACOS device. (With L2 DSR, the ACOS device must be on the same subnet as the real servers. With L3 DSR, the device and real servers can be separated by a router.)

The packets from the real servers must appear as though they are actually coming from the ACOS VIP and *not* from the real servers. Using IP-in-IP Tunneling helps make this happen.

Requirements

This IP-in-IP Tunneling for L3 DSR configuration requirement are as follows:

Requirements on the ACOS device:

- In order to preserve the original client IP address, (which will be used as the destination address for packets sent from the real server to the client), make sure not to enable Source NAT under the virtual port where the `ipinip` command is used.
- L3 DSR is supported on the following virtual port types (service types): TCP, UDP, FTP, and TFTP for both IPv4 and IPv6 protocols.

Requirements on the real server:

- A loopback interface must be configured with the virtual server IP address. (If this does not work on your real server, it may be helpful to configure the real server end of the IP Tunnel with the ACOS VIP address.)

- ARP replies from the loopback interfaces must be disabled. (This applies to the loopback interfaces that have the virtual server IP address.)
- The real server must be configured with an IP Tunnel, and it must be configured to decapsulate traffic received from the ACOS device over that tunnel.

Methods to Display the ACOS VIP as the Source Address

In previous releases, the ACOS device supports L3 DSR by using the Differentiated Services Code Point (DSCP) field. This field is not used to support QoS (as per its original intention), instead used to create a mapping between the DSCP value and the VIPs on the ACOS device. This information is then stored in a table on the real server, and the server uses the table as a reference when it needs to know which VIP to include as the Source Address in the packet it sends back to the client.

Using the DSCP value to encode VIP-mappings may not work in all network environments. For example, the feature is not supported on Windows-based servers.

This release introduces a new mechanism that can be used to get the VIP to appear as the Source Address in the packet that the real server sends back to the client. This new approach is called IP-in-IP Tunneling, and it provides a helpful alternative for L3 DSR deployments in which DSCP is not supported.

IP-in-IP Tunneling for L3 DSR Routing

IP-in-IP Tunneling, or just “IP tunneling”, is commonly used to connect networks that use incompatible protocols. When traffic from a source network arrives at a network that uses an incompatible protocol (such as IPv4 and IPv6), the IP tunneling protocol can be used to add an outer header to the original packet, thus encapsulating the original packet in another packet.

Through this process of encapsulating the original packet, traffic can be reliably carried from the source network, across a dissimilar transit network, and back to a network that uses the same protocol as the source network. When the packet reaches the far side of the IP tunnel, the outer header is stripped off (decapsulated), and the original packet arrives intact.

When IP Tunneling for L3 DSR is enabled on the ACOS device, an extra IP header is added to the client packet before it is forwarded over the IP tunnel to a real server.

These real servers are configured to strip off the extra IP encapsulation layer, yielding the original packet sent by the client. This packet is intact and otherwise unaffected by the encapsulation process.

By removing the outer header, the real server now has a packet with the source address of the client and the destination address of the ACOS VIP. (This would not have been true if the ACOS device had used standard Source NAT and Destination NAT to process the packet.) When the real server responds to the client, it crafts its response based upon a “pristine” packet that has not had its source and destination addresses modified by the NAT process.

When the real server responds to the client, source and destination addresses are reversed:

- The packet’s source address (originally the client IP) is changed to become the ACOS VIP.
- The packet’s destination address (originally the ACOS VIP) is changed to become the client’s IP.

IP-in-IP Tunneling for L3 DSR Routing Details include:

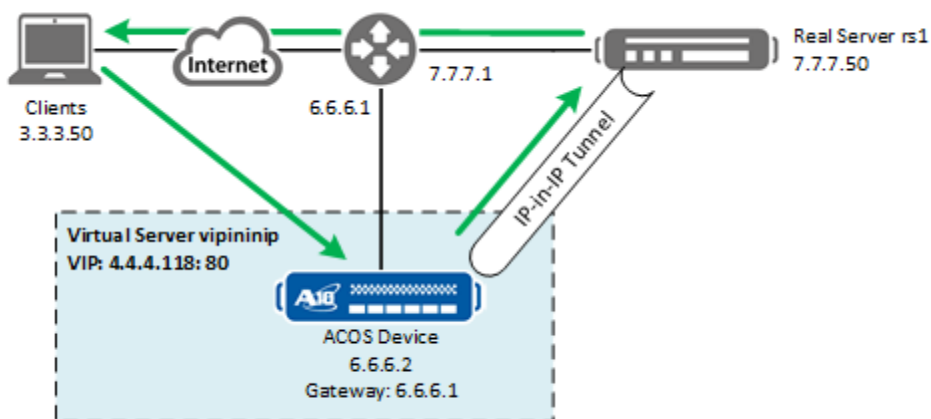
- Both IPv4 and IPv6 IP in IP are supported.
- Whereas using the DSCP value was only supported on Linux servers, IP tunneling for L3 DSR can be set up on most popular brands of web servers, such as Linux, Unix, or Windows.
- IP tunneling must be configured on each real server for which IP Tunneling for L3 DSR is enabled. Further, each real server must be configured to decapsulate the packets it receives and send those packets to the client that originated the request.
- In Direct Server Return (DSR) configurations, member servers in a Service Group must be listening on the same port. Port translation is not supported in DSR topologies.
- For more information about IP in IP (or "ipencap"), refer to RFC 2003.

[Figure 13](#) shows an ACOS device deployed in an L3 Direct Server Return (DSR) configuration. In DSR configurations, replies from real servers do not pass through the ACOS device, returning directly to the client.

In this example, the ACOS device is attached to the network in a “one-armed” configuration, through an Ethernet port or a trunk. (Our configuration sample shown later in this section uses Ethernet port 3.)

The green arrows show the traffic flow for the traffic from the client at 3.3.3.50 and the real server at 7.7.7.50. Client requests go to the virtual server IP address at 4.4.4.118 (port 80). The ACOS device encapsulates the packets in an IP Tunnel and then forwards them to real server. The real server de-capsulates the traffic and processes the client’s original packet. However, instead of sending the traffic back to the ACOS device, the real server sends the traffic directly to the client, bypassing the ACOS device.

Figure 13 : ACOS Deployment Example – IP Tunneling for DSR



DSR Health Checking

It is recommended that you configure Layer 3 or Layer 4-7 health checks, both of which are supported in DSR configurations. When using DSR health checking, a separate service group is required for each individual virtual server.

Layer 3 DSR Health Checks

The target of the Layer 3 health checks can be the real IP addresses of the servers, or the virtual IP address, depending on your preference.

- To send the Layer 3 health checks to the real server IP addresses, you can use the default Layer 3 health method (ICMP).
- To send the Layer 3 health checks to the virtual IP address instead:
 - Configure an ICMP health method with transparent option enabled and alias address set to the virtual IP address.
 - Globally enable DSR health checking.

Layer 4-7 DSR Health Checks

Layer 4-7 health checks are sent to the same IP address as the Layer 3 health checks, and then addressed to the specific protocol port. You can use the default TCP and UDP health monitors or configure new health monitors. This example uses the default TCP health monitor.

When IP tunneling is enabled, health check packets sent to the server are encapsulated when sent through the IP tunnel.

CLI Configuration

This topic describes CLI configuration example of IP tunneling for DSR.

Enable an Ethernet interface

```
ACOS(config)# interface ethernet 3
ACOS(config-if:ethernet:3)# ip address 6.6.6.2 /24
ACOS(config-if:ethernet:3)# enable
ACOS(config-if:ethernet:3)# exit
```

Configure the Health Checks

```
ACOS(config)# health monitor hm-http
ACOS(config-health:monitor)# method http
ACOS(config-health:monitor)# exit
```

Globally Enable DSR Health Checking of VIPs

```
ACOS(config)# slb common
ACOS(config-common)# dsr-health-check-enable
ACOS(config-common)# exit
```

Configure the Real Servers


```
ACOS(config)# slb server rs1 7.7.7.50
ACOS(config-real server)# port 80 tcp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
```

Configure the Service Group

For this configuration, the health monitor must be configured at the service group configuration level. If you try to apply the health monitor at the server port configuration level, the port may not come up.

```
ACOS(config)# slb service-group sg1 tcp
ACOS(config-slb svc group)# member rs1 80
ACOS(config-slb svc group-member:80)# exit
ACOS(config-slb svc group)# health-check hm-http
ACOS(config-slb svc group)# exit
```

Configure the Virtual Server with 'ipinip' Option

```
ACOS(config)# slb virtual-server vipipinip 4.4.4.118
ACOS(config-slb vserver)# port 80 tcp
ACOS(config-slb vserver-vport)# service-group sg1
ACOS(config-slb vserver-vport)# ipinip
ACOS(config-slb vserver-vport)# exit
ACOS(config-slb vserver)# exit
```

Configure the Virtual Server with Javascript CSS

```
ACOS(config)# slb virtual-server vipipinip 4.4.4.118
ACOS(config-slb vserver)# port 80 tcp
ACOS(config-slb vserver-vport)#jsi-enable file-inspect
```

where **jsi-enable** tag can be one of the following - file, inspect, ssl, slb, waf. Currently only file-inspect option is supported.

Configuration on the Real Servers

For the IP-in-IP Tunneling for L3 DSR feature to work correctly, the following configurations must be performed on each real server that will be process traffic from the IP tunnel:

- Configure a VIP on the real server and disable ARP for it. This VIP is typically configured as a loopback interface.
- Configure an IP-in-IP tunnel on each real server.

When properly configured, the real server decapsulates packets received from the IP tunnel, processes requests in the client's original packet, and sends responses directly to the client bypassing the ACOS device.

Route Health Injection Deployment

This section provides deployment examples for Route Health Injection (RHI) for Server Load Balancing (SLB).

The following topics are covered:

Overview	96
Deployment Examples	97
Configuration Considerations	99
Basic RHI with BGP	100
Basic RHI with OSPF	109
Advanced RHI with OSPF for VRRP-A	112

Overview

Route Health Injection (RHI) allows you to distribute a healthy virtual IP (VIP) address that represents an application as a host route (/32) using standards-based protocols like OSPF and BGP-4. Using your existing routing infrastructure, this type of configuration deployment enables the application IP address to appear in multiple locations.

Why Use Health Injection

RHI provides a means of connecting to an application IP address (See 10.4.1.1 in [Figure 14](#)), which resides in more than one location. This is useful for providing Multi-Site Load Balancing, Disaster Recover, and Redundancy without having to deploy DNS based Global Load Balancing (GSLB). This can also be useful for applications that cannot dynamically learn another IP address for redundancy and can only use a single address once learned. RHI requires a Host Route, which is represented by the entire 32-bit IP address without any summarization (255.255.255.255 or /32) in the routing table. This route is injected into the routing infrastructure based on the health of the application the VIP IP address is representing. Traditionally a route in the router represents a subnet and not an individual host. However, a router will look for the best route to a device or subnet and will forward the packet to the next hop of the longest prefix match, in this case the host route. The router is not limited to the length of a route prefix and a host route is a legitimate route.

Custom Host Route

Thunder platform allows customization of the host route to be injected with a custom metric on a site-by-site basis providing control of the preferred route to the application when the router makes a routing decision. This allows for the traffic to be directed based on user preference to the preferred application server or site.

For instance, if a site always has a lower priority (Lower is preferred) than a disaster recovery site, the primary site will always be selected unless the route is withdrawn. The second route to the Disaster Recovery site will then become the best route and the traffic will then directed to the Disaster Recovery site. Another aspect of RHI is determining when to inject or withdraw the VIP Route. Thunder platform uses a variety of customizable health check methods to determine if the

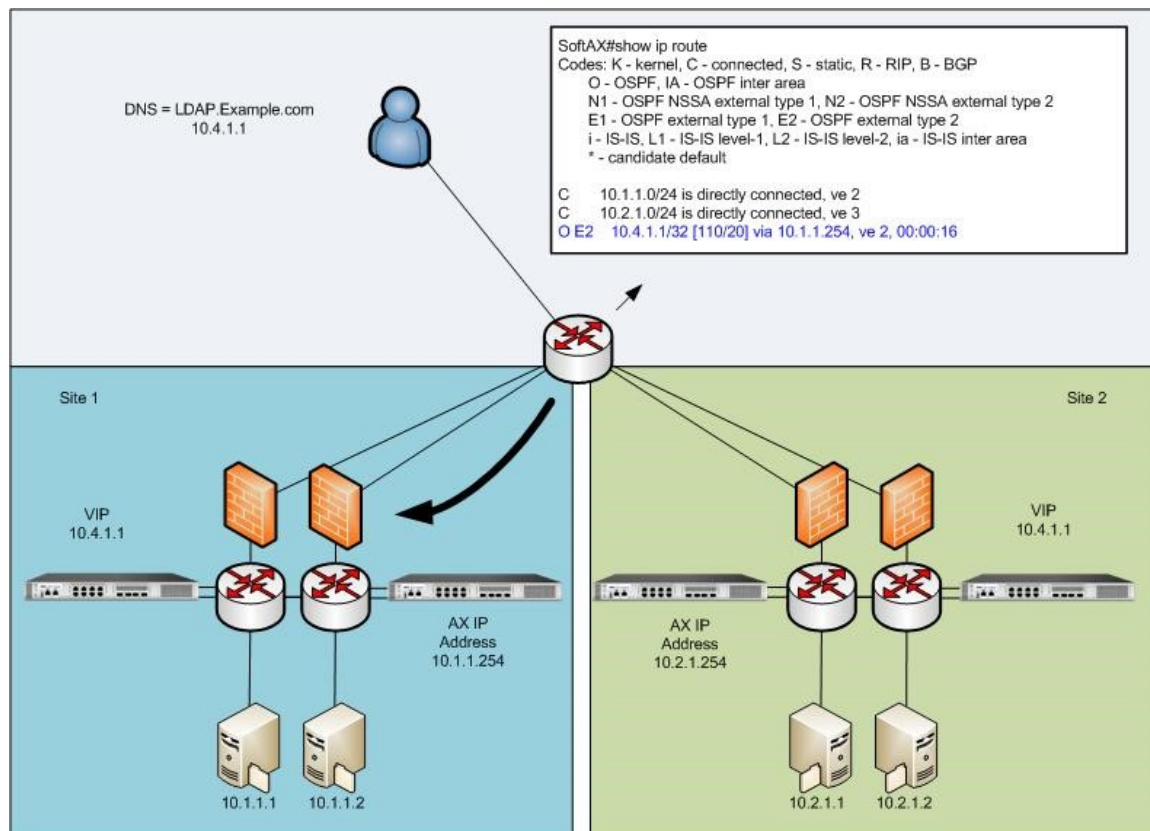
local Application is healthy and reachable. This information can be used to make decision to inject the VIP Route if the application is healthy, or to withdraw the route if the application is failing its health check using routing protocols.

Deployment Examples

Site Preference

The below is an example of RHI being used to provide Disaster Recover or site preference for all traffic to an LDAP server application at multiple sites. Disaster Recovery is accomplished by the site Thunder's injecting a VIP route (10.4.1.1) into the routing table with different metrics. Site-1 is injecting the route with a metric of 20 and Site-2 is injecting the route with a metric of 30. Looking at the central router's routing table with a "Show IP Route", the route below highlighted in blue is the preferred route to reach 10.4.1.1 because the route received from 10.1.1.254 (Site-1) has a metric of 20 which is lower cost than 30. The lower metric is preferred. Both routes exist in the OSPF database, but only the best route is populated into the routing table.

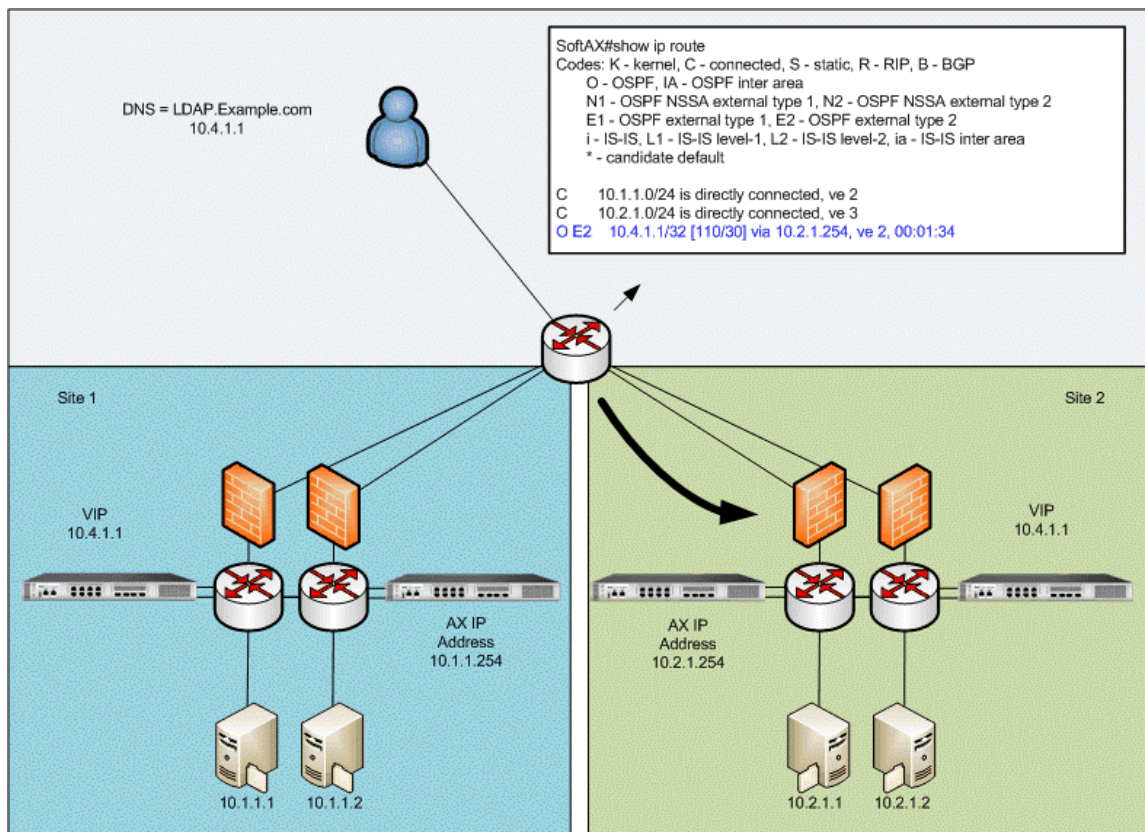
Figure 14 : ACOS Deployment Example – Disaster Recovery for Site Preference



Backup Site

If a failure of the Site-1 occurs, the remaining route listed in below in blue, will now become the most preferred route, and traffic will be directed to the Disaster Recovery or back up site. This can be a dynamic or manual process. A route can simply be injected by enabling a VIP address that is in waiting, or by adjusting the metric to be preferred. This in most cases is dynamic based on the VIP health determined by the health check defined to monitor the real application.

Figure 15 : ACOS Deployment Example – Disaster Recovery for Site Backup



Configuration Considerations

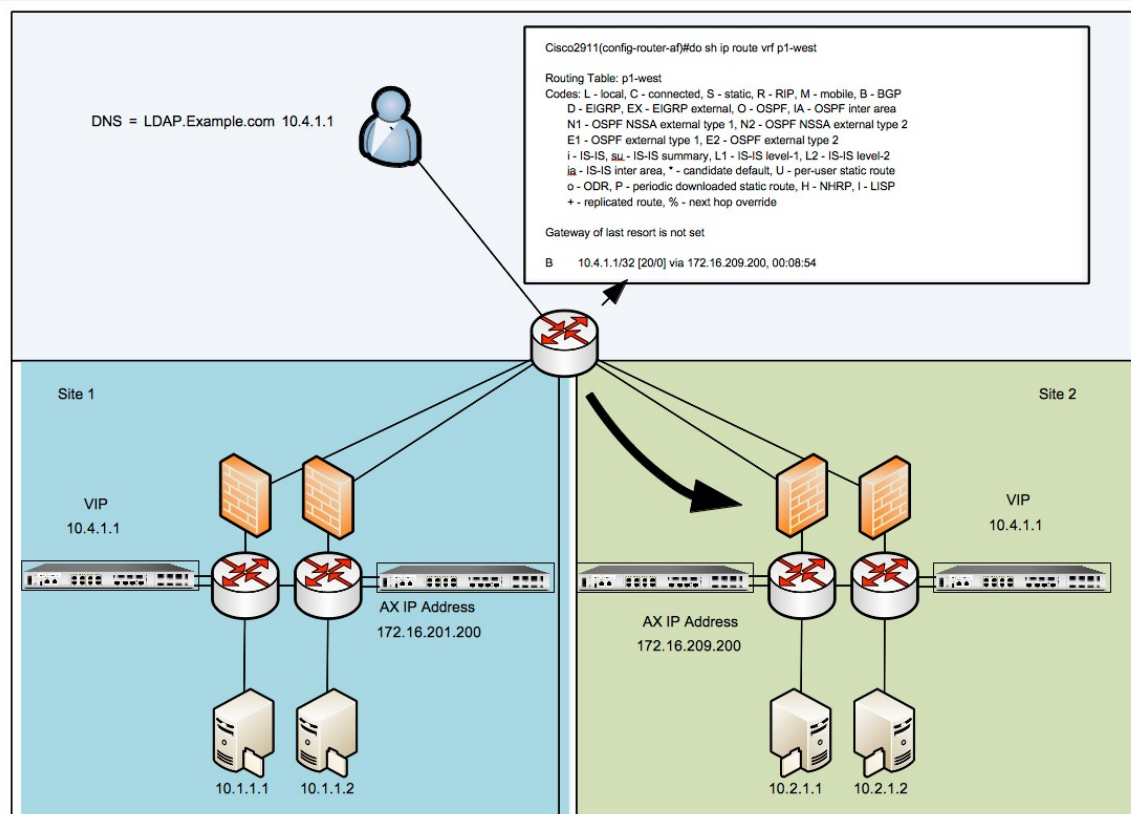
Before you enable RHI, consider the following:

- *Requires the router to accept Host Routes.* Most service providers will not allow a /32 route to be received into their BGP Network. Therefore, an agreement needs to be reached with your service provider to accept Host Routes if a Service Provider provides connectivity between sites. This is not an issue with private networks.
- *Redistributing Routes into multiple routing protocols.* The timing of the failover (When the new route becomes preferred) can be based on routes timing out of the routing table.

Basic RHI with BGP

This topic describes the basic RHI with BGP. The following figure is the deployment example.

Figure 16 : Basic RHI with BGP



It is required to redistribute the host route for the VIP into the routing protocol. To enable redistribution, enter the following commands to enable BGP and apply redistribution to the VIP. This will inject a /32 route for every flagged VIP from the active box. If the ACOS device were to go into standby, route would be withdrawn, and active ACOS device would advertise the route. A router-id needs to be unique per ACOS devices.

ACOS Device-1 Configuration

```
ACOS(config)# route-map example1 permit 10
ACOS(config-route-map:10)# set metric 50
```



```
ACOS(config)# router bgp 65529
ACOS(config-bgp:65529)# bgp router-id 172.16.201.200
ACOS(config-bgp:65529)# bgp log-neighbor-changes
ACOS(config-bgp:65529)# neighbor 172.16.201.1 remote-as 2
ACOS(config-bgp:65529)# redistribute vip only-flagged route-map example1
```

ACOS Device-2 Configuration

```
ACOS(config)# route-map example1 permit 10
ACOS(config-route-map:10)# set metric 60

ACOS(config)# router bgp 65529
ACOS(config-bgp:65529)# bgp router-id 172.16.209.200
ACOS(config-bgp:65529)# bgp log-neighbor-changes
ACOS(config-bgp:65529)# neighbor 172.16.209.1 remote-as 2
ACOS(config-bgp:65529)# redistribute vip only-flagged route-map example1
```

Enable redistribution of a flagged application VIP.

```
ACOS(config)# slb virtual-server vip1 10.4.1.1
ACOS(config-slb vserver)# redistribution-flagged
ACOS(config-slb vserver)# arp-disable
ACOS(config-slb vserver)# port 80 http
ACOS(config-slb vserver-vport)# name _10.4.1.1_HTTP_80
ACOS(config-slb vserver-vport)# aflex localserver
```

Configuring Route Map

To enable redistribution of an injected VIP route into BGP with a specific next-hop route specified as the VRRP-A Floating IP address follow the below steps:

1. Create a route map to set the desired next hop for redistributed VIP routes.

The route-map (example2) defines the next hop that BGP will use for the redistributed VIPs.

```
ACOS(config)# route-map example2 permit 20
ACOS(config-route-map:20)# set ip next-hop 10.1.1.252
```

2. Apply the route-map to the BGP redistribution statement. This tells BGP to use the above next-hop when redistributing VIP routes.

```
ACOS(config)# router bgp 65529
ACOS(config-bgp:65529)# bgp log-neighbor changes
ACOS(config-bgp:65529)# bgp router-id 10.1.1.253
ACOS(config-bgp:65529)# neighbor 10.1.1.251 remote-as 65529
ACOS(config-bgp:65529)# redistribute vip only-flagged route-map
example2
```

3. Enable redistribution of a flagged application VIP.

```
ACOS(config)# slb virtual-server vip1 10.4.1.1
ACOS(config-slb vserver)# disable-when-all-ports-down
ACOS(config-slb vserver)# redistribution-flagged
ACOS(config-slb vserver)# port 80 tcp
ACOS(config-slb vserver-vport)# source-nat pool SNAT-Pool1
ACOS(config-slb vserver-vport)# service-group web
```

Consolidated Configuration Example

Device 1 Configuration:



```
vlan 2
  untagged ethernet 1
  router-interface ve 2
!
hostname ACOS-1
!
interface management
  ip address 10.64.19.75 255.255.255.0
  ip default-gateway 10.64.19.1
!
interface ve 2
  ip address 172.16.201.200 255.255.255.0
!
ip nat pool SNAT-Pool1 10.1.1.10 10.1.1.10 netmask /24
!
slb server real1 10.1.1.1
  port 80 tcp
!
slb server real2 10.1.1.2
  port 80 tcp
!
slb service-group web tcp
  member real1 80
  member real2 80
!
slb virtual-server vip1 10.4.1.1
  redistribution-flagged
  port 80 tcp
  source-nat pool SNAT-Pool1
  service-group web
!
router bgp 65529
  bgp log-neighbor-changes
  bgp router-id 172.16.201.200
  neighbor 172.16.201.1 remote-as 2
  redistribute vip only-flagged route-map example1
!
route-map example1 permit 10
```

```
set metric 50
!  
!  
end
```

Device 2 Configuration:



```
hostname ACOS-2
!
vlan 2
untagged ethernet 1
router-interface ve 2
!
interface management
  ip address 10.64.19.76 255.255.255.0
  ip default-gateway 10.64.19.1
!
interface ve 2
  ip address 172.16.209.200 255.255.255.0
!
ip nat pool SNAT-Pool1 10.2.1.10 10.2.1.10 netmask /24
!
slb server real1 10.2.1.1
  port 80 tcp
!
slb server real2 10.2.1.2
  port 80 tcp
!
slb service-group web tcp
member real1_80
member real2_80
!
slb virtual-server VIP-1 10.4.1.1
  disable-when-all-ports-down
  redistribution-flagged
  port 80 tcp
  source-nat pool SNAT-Pool1
  service-group web
router bgp 65529
  bgp router-id 172.16.209.200
  bgp log-neighbor-changes
  neighbor 172.16.209.1 remote-as 2
  redistribute vip only-flagged example1
!
route-map example1 permit 10
```



```
set metric 60
!  
!  
end
```

Basic RHI with OSPF

This topic describes the basic RHI with OSPF configuration example.

It is required to redistribute the host route for the VIP into the routing protocol. To enable redistribution of the VIP into OSPF with a specific metric enter the following commands to enable OSPF and redistribution of a healthy VIP. This will inject a /32 route for every flagged VIP with a route metric of 20.

```
ACOS(config)# router ospf  
ACOS(config)# ospf router-id 10.1.1.254  
ACOS(config)# redistribute vip only-flagged metric 20  
ACOS(config)# network 10.1.1.0 0.0.0.255 area 0
```

Enable redistribution of a flagged application VIP.

```
ACOS(config)# slb virtual-server LDAP-VIP 10.4.1.1  
ACOS(config)# redistribution-flagged  
ACOS(config)# port 80 tcp  
ACOS(config)# source-nat pool SNAT-Pool1  
ACOS(config)# service-group web
```

Consolidated Configuration Example

Device 1 Configuration

```
hostname ACOS-1  
!  
clock timezone America/Chicago  
!  
vlan 2  
  untagged ethernet 1  
  router-interface ve 2  
!
```

```
interface management
 ip address 172.31.31.31 255.255.255.0
!
interface ve 2
 ip address 10.1.1.254 255.255.255.0
 ip ospf cost 1
!
ip route 0.0.0.0 /0 10.1.1.253
!
router ospf
 ospf router-id 10.1.1.254
 redistribute vip only-flagged metric 20
 network 10.1.1.0 0.0.0.255 area 0
!
ip nat pool SNAT-Pool1 10.1.1.10 10.1.1.10 netmask /24
!
slb server real1 10.1.1.1
 port 389 tcp
!
slb server real2 10.1.1.2
 port 389 tcp
!
slb service-group web tcp
 member real1_389
 member real2_389
!
slb virtual-server LDAP-VIP 10.4.1.1
 redistribution-flagged
 port 80 tcp
 source-nat pool SNAT-Pool1
 service-group web
!
enable-management service ssh ve 2
enable-management service telnet ve 2
enable-management service http ve 2
enable-management service https ve 2
enable-management service snmp ve 2
!
end
```

Device 2 Configuration

```
hostname ACOS-2
!
clock timezone America/Chicago
!
vlan 2
  untagged ethernet 1
  router-interface ve 2
!
interface management
  ip address 172.31.31.31 255.255.255.0
!
interface ve 2
  ip address 10.2.1.254 255.255.255.0
  ip ospf cost 1
!
ip route 0.0.0.0 /0 10.2.1.253
!
router ospf
  ospf router-id 10.2.1.254
  redistribute vip only-flagged metric 30
  network 10.2.1.0 0.0.0.255 area 0
!
ip nat pool SNAT-Pool1 10.2.1.10 10.2.1.10 netmask /24
!
slb server real1 10.2.1.1
  port 389 tcp
!
slb server real2 10.2.1.2
  port 389 tcp
!
slb service-group web tcp
  member real1_389
  member real2_389
!
slb virtual-server LDAP-VIP 10.4.1.1
  redistribution-flagged
  port 80 tcp
  source-nat pool SNAT-Pool1
  service-group web
```

```
!  
enable-management service ssh ve 2  
enable-management service telnet ve 2  
enable-management service http ve 2  
enable-management service https ve 2  
enable-management service snmp ve 2  
!  
end
```

Advanced RHI with OSPF for VRRP-A

This topic describes the advanced RHI with OSPF in the VRRP-A environment.

It is essential to configure the system to establish peering or neighboring most efficiently to avoid packet loss. You can achieve this by using a floating address or a VRRP-A address as the next-hop for the injected VIP routes. This allows traffic to learn the new path to the VIP route quickly.

In addition, when the VRRP-A floating address moves from the failed active device to the standby device, the mechanism provides immediate failover from the active to the standby unit. The address quickly moves from active to the standby when failover happens, and the new active unit sends out gratuitous ARPs to force all switches and routers to rapidly learn the new location of the IP and MAC address.

To create an OSPF Route that uses the address of the next hop floating address, you need to create a route map and specify the appropriate address information that will apply various metrics for use in OSPF or BGP.

A route map can also provide multiple options to control traffic to and from an injected VIP route and even multiple VIP routes. For instance, you can set different route metrics per VIP IP address, allowing you to have one site active for one application and another site active for a completely different application.

Configuring Route map

To enable redistribution of an injected VIP route into OSPF with a specific next-hop route specified as the floating HA address or a VRRP-A address, follow the below steps.

1. Create a route map and specify the desired next hop.

```
ACOS(config)# router ospf
ACOS(config)# ospf router-id 10.1.1.254
ACOS(config)# redistribute vip only-flagged route-map example1 metric
20
ACOS(config)# network 10.1.1.0 0.0.0.255 area 0
```

2. Create a route map to set the next hop that will be used to reach the VIP address, and specify the route map in the OSPF redistribution statement.

NOTE: This is a global setting for all VIPs.

```
ACOS(config)# route-map example1 permit 20
ACOS(config)# set ip next-hop 10.1.1.251
```

3. Enable redistribution of a flagged application VIP.

```
ACOS(config)# slb virtual-server LDAP-VIP 10.4.1.1
ACOS(config)# disable when-all-ports-down
ACOS(config)# redistribution-flagged
ACOS(config)# port 80 tcp
ACOS(config)# source-nat pool SNAT-Pool1
ACOS(config)# service-group web
```

Configuring Prefix list within Route Map

To set route parameters for an individual VIP, you must create a prefix list matching the injected VIP route, follow the below steps.

1. Create the route map for the injected VIP route.

```
ACOS(config)# router ospf
ACOS(config)# ospf router-id 10.1.1.254
ACOS(config)# redistribute vip only-flagged route-map example1
ACOS(config)# network 10.1.1.0 0.0.0.255 area 0
```

2. Create a prefix list to match VIP IP address 1.

```
ACOS(config)# ip prefix-list MatchVIP1 seq 5 permit 10.4.1.1/32
```

3. Create a prefix list to match VIP IP address 2.

```
ACOS(config)# ip prefix-list MatchVIP2 seq 5 permit 10.4.1.2/32
```

4. Create route maps to set the next hop that will be used to reach the VIP address, and specify the route map in the OSPF redistribution statement. These settings are for each individual VIP.
5. Create a route map and set the parameters for the VIP IP address specified in prefix-list 1.

```
ACOS(config)# route-map example1 permit 20
ACOS(config)# match ip address prefix-list MatchVIP1
ACOS(config)# set metric 100
ACOS(config)# set ip next-hop 10.1.1.251
```

6. Create an entry in the same route-map with an incremental sequence and set the parameters for the VIP IP address specified in prefix-list 2.

If traffic matches prefix-list 1 it will be executed first, similar to the execution of an access list, so the prefix needs to be specific to the VIP IP Addresses.

```
ACOS(config)# route-map example1 permit 30
ACOS(config)# match ip address prefix-list MatchVIP2
ACOS(config)# set metric 200
ACOS(config)# set ip next-hop 10.1.1.251
```

7. Enable redistribution of a flagged application VIP.

```
ACOS(config)# slb virtual-server LDAP-VIP 10.4.1.1
ACOS(config)# disable when-all-ports-down
ACOS(config)# redistribution-flagged
ACOS(config)# port 80 tcp
ACOS(config)# source-nat pool SNAT-Pool1
ACOS(config)# service-group web
```

Consolidated Configuration Example

Consolidated configuration for OSPF Route Health Injection in VRRP-A environment

Device 1 Configuration:

```
hostname Site1-ACOS Device 1
!
vrrp-a common
```

```
device-id 1
set-id 1
enable
!
vlan 2
  untagged ethernet 1
  router-interface ve 2
!
interface ve 2
  ip address 10.1.1.254 255.255.255.0
  ip ospf cost 1
!
ip route 0.0.0.0 /0 10.1.1.250
!
router ospf
  ospf router-id 10.1.1.254
  redistribute vip only-flagged route-map example1
  network 10.1.1.0 0.0.0.255 area 0
!
ip prefix-list 1 seq 5 permit 10.4.1.1/32
ip prefix-list 2 seq 5 permit 10.4.1.2/32
!
route-map example1 permit 20
  match ip address prefix-list 1
  set metric 100
  set ip next-hop 10.1.1.251
!
route-map example1 permit 30
  match ip address prefix-list 2
  set metric 200
  set ip next-hop 10.1.1.251
!
vrrp-a vrid 0
  floating-ip 10.1.1.251
  blade-parameters
  priority 200
!
ip nat pool SNAT-Pool1 10.1.1.10 10.1.1.10 netmask /24
!
vrrp-a interface ethernet 1
```

```
!  
slb server real1 10.1.1.1  
    port 80 tcp  
!  
slb server real2 10.1.1.2  
    port 80 tcp  
!  
slb service-group web tcp  
    member real1_80  
    member real2_80  
!  
slb virtual-server VIP-1 10.4.1.1  
    disable when-all-ports-down  
    redistribution-flagged  
    port 80 tcp  
    source-nat pool SNAT-Pool1  
    service-group web  
!  
slb virtual-server VIP-1 10.4.1.2  
    disable when-all-ports-down  
    redistribution-flagged  
    port 80 tcp  
    source-nat pool SNAT-Pool1  
    service-group web
```

Device 2 Configuration:

```
hostname Site1-ACOS Device 2  
!  
vrrp-a common  
    device-id 2  
    set-id 1  
    enable  
!  
vlan 2  
    untagged ethernet 1  
    router-interface ve 2  
!  
interface ve 2  
    ip address 10.1.1.253 255.255.255.0
```



```
ip ospf cost 1
!
ip route 0.0.0.0 /0 10.1.1.250
!
router ospf
  ospf router-id 10.1.1.253
  redistribute vip only-flagged route-map example1
  network 10.1.1.0 0.0.0.255 area 0
!
ip prefix-list 1 seq 5 permit 10.4.1.1/32
ip prefix-list 2 seq 5 permit 10.4.1.2/32
!
route-map example1 permit 20
  match ip address prefix-list 1
  set metric 100
  set ip next-hop 10.1.1.251
!
route-map example1 permit 30
  match ip address prefix-list 2
  set metric 200
  set ip next-hop 10.1.1.251
!
vrrp-a vrid 0
  floating-ip 10.1.1.251
!
ip nat pool SNAT-Pool1 10.1.1.10 10.1.1.10 netmask /24
!
vrrp-a interface ethernet 1
!
slb server real1 10.1.1.1
  port 80 tcp
!
slb server real2 10.1.1.2
  port 80 tcp
!
slb service-group web tcp
  member real1_80
  member real2_80
!
slb virtual-server VIP-1 10.4.1.1
```

```
disable when-all-ports-down
redistribution-flagged
port 80 tcp
source-nat pool SNAT-Pool1
service-group web
!
slb virtual-server VIP-1 10.4.1.2
disable when-all-ports-down
redistribution-flagged
port 80 tcp
source-nat pool SNAT-Pool1
service-group web
```

Network Address Translation

The following topics are covered:

Overview	120
Direct Server Return	124
IP NAT Support for VIPs	125
Using IP Pool Default Gateways	127
Smart NAT for Virtual Ports	128
Virtual-port TCP Maximum Segment Life	133
Non-standard Port Support	134

Overview

This section describes Network Address Translation (NAT) and how to configure it. NAT translates the source or destination IP address of a packet before forwarding the packet.

ACOS uses NAT to perform SLB.

NOTE: This section does not include information about Carrier Grade NAT (CGN) or other NAT features for IPv6 migration.

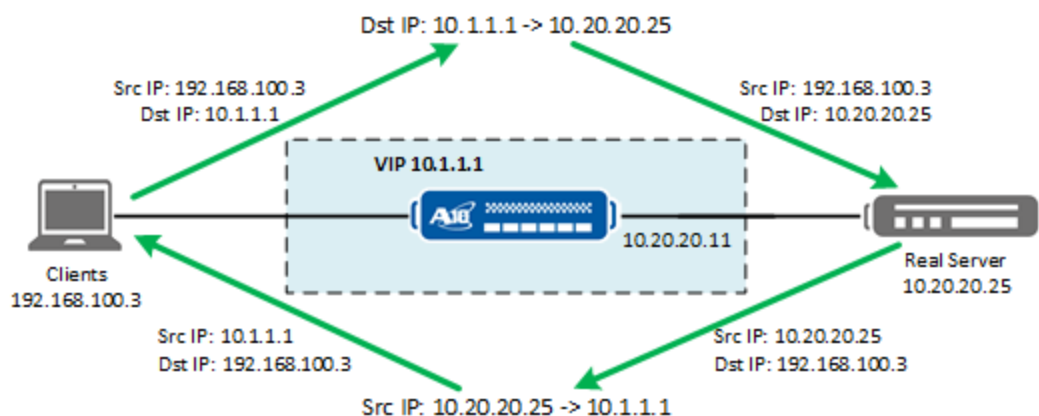
ACOS devices can perform source and destination NAT on client-VIP SLB traffic.

SLB Destination NAT

ACOS devices automatically perform destination NAT for client-VIP SLB traffic. [Figure 17](#) shows an example.

Destination NAT is disabled for virtual ports on which Direct Server Return (DSR) is enabled.

Figure 17 : SLB NAT



By default, SLB NAT works as follows.

- Before forwarding a client packet to a real server, the ACOS device translates the destination IP address from the virtual server IP address (VIP) to the IP address of the real server.
- ACOS reverses the translation before sending the server reply to the client. The source IP address is translated from the real server's IP address to the VIP address.

The default SLB NAT behavior does not translate the client's IP address.

SLB Source NAT

SLB source NAT is disabled by default. There are some cases where SLB Source NAT is applicable:

- Connection reuse. (See [Connection Reuse](#).)
- The VIP and real servers are in different subnets. In cases where real servers are in a different subnet than the VIP, source NAT ensures that reply traffic from a server will pass back through the ACOS device. (See [Source NAT for Servers in Other Subnets](#).)

Connection Reuse

Connection reuse enables you to reuse TCP connections between the ACOS device and real servers for multiple client sessions. When you enable this feature, the ACOS device does not tear down a TCP connection with the real server each time a client ends its session. Instead, the ACOS device leaves the TCP connection established, and reuses the connection for the next client that uses the real server.

Since the TCP connection with the real server needs to remain established after a client's session ends, the client's IP address cannot be used as the source address for the connection. Instead, the source address must be an IP address from a NAT pool or pool group configured on the ACOS device.

Prerequisite

- Connection reuse requires SLB source NAT.

The following topics are covered:

Source NAT for Servers in Other Subnets

ACOS allows source NAT to be enabled on a virtual port. In cases where real servers are in a different subnet than the VIP, source NAT ensures that reply traffic from a server will pass back through the ACOS device.

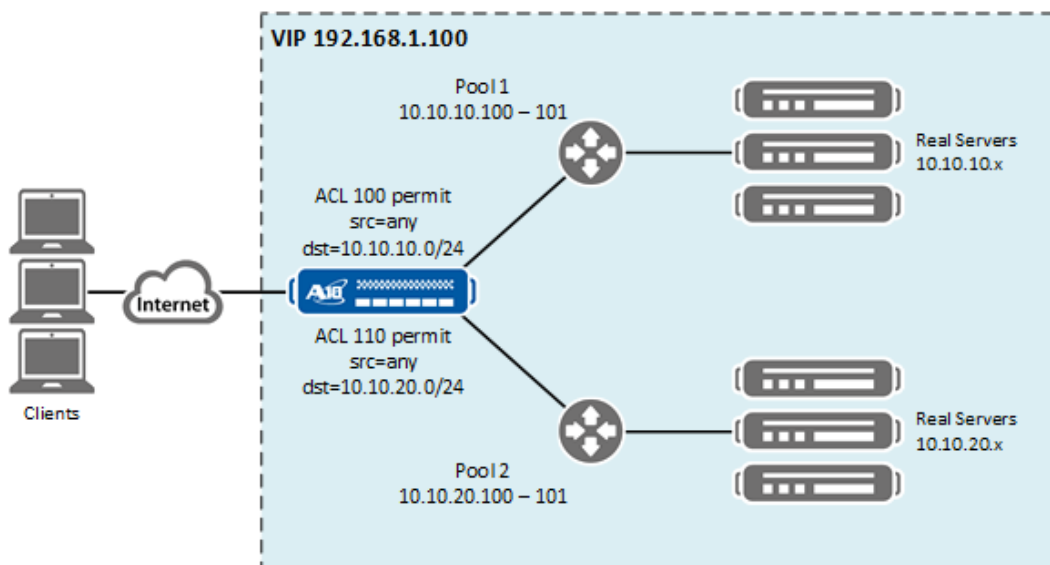
You can enable source NAT on a virtual port in either of the following ways:

- Use the source-nat option to bind a single IP address pool or pool group to the virtual port. This option is applicable if all the real servers are in the same subnet.
- Use sets of ACL-pool pairs, one for each real server subnet. You must use this method if the real servers are in multiple subnets. This section describes how to use this method.

For the real server to be able to send replies back through the ACOS device, use an extended ACL. The source IP address must match on the client address. The destination IP address must match on the real server address. The action must be permit.

The ACL should not match on the virtual IP address (unless the virtual IP address is in the same subnet as the real servers, in which case source NAT is probably not required). [Figure 18](#) shows an example.

Figure 18 : Multiple NAT Pools Bound to a Virtual Port



In this example, a service group has real servers that are located in two different subnets. The VIP is not in either of the subnets. To ensure that reply traffic from a server will pass back through the ACOS device, the ACOS device uses IP source NAT.

To implement IP source NAT, two pairs of ACL and IP address pool are bound to the virtual port. Each ACL-pool pair contains the following:

- An extended ACL whose source IP address matches on client addresses and whose destination IP address matches on the real server's subnet.
- An IP address pool or pool group containing translation addresses in the real server's subnet.

For example, if SLB selects a real server in the 10.10.10.x subnet, then the source IP address is translated from the client's address to an address in pool 1. When the server replies, it replies to the address from pool 1.

In most cases, destination NAT does not need to be configured for SLB. ACOS automatically translates the VIP address into a real server address before forwarding a request to the server.

Source NAT for Servers in Other Subnets (CLI Example)

The following commands implement the source NAT configuration shown in [Figure 18](#).

First, the ACLs are configured. In each ACL, "any" is used to match on all clients. The destination address is the subnet where the real servers are located.

```
ACOS(config)# access-list 100 permit ip any 10.10.10.0 /24
ACOS(config)# access-list 110 permit ip any 10.10.20.0 /24
```

The following commands configure the IP address pools. Each pool contains addresses in one of the real server subnets.

```
ACOS(config)# ip nat pool pool1 10.10.10.100 10.10.10.101 netmask /24
ACOS(config)# ip nat pool pool2 10.10.20.100 10.10.20.101 netmask /24
```

The following commands bind the ACLs and IP address pools to a virtual port on the VIP:

```
ACOS(config)# slb virtual-server vip1 192.168.1.100
ACOS(config-slb vserver)# port 80 tcp
ACOS(config-slb vserver-vport)# access-list 100 source-nat-pool pool1
```

```
ACOS(config-slb vserver-vport)# access-list 110 source-nat-pool pool2
```

Direct Server Return

You can disable destination NAT on a virtual port, to enable Direct Server Return (DSR). DSR enables a real server to respond to clients directly instead of going through the ACOS device. The ACOS is not required to return the server's response traffic to clients, so there is no need to un-NAT traffic.

This type of NAT is especially useful for applications that have intensive payload transfers, such as FTP and streaming media.

When DSR is enabled, only the destination MAC address is translated from the VIP's MAC address to the real server's MAC address. The destination IP address is still the VIP.

To use DSR, the ACOS device and the real servers must be in the same Layer 2 subnet. The VIP address must be configured as a loopback address on the real servers.

To enable DSR on a virtual port, use the following method(s).

The current release does not support external health monitoring for DSR deployments. To configure health checking for DSR, see [Health Monitoring Virtual IP Addresses in DSR Deployments](#).

For examples of DSR configurations, see [Direct Server Return \(DSR\) SLB Deployment](#).

GUI Configuration

1. Navigate to **ADC >>SLB >> Virtual Servers** tab.
2. Select the virtual server name of an existing virtual server, or click **Create** to add a new virtual server.
3. If you are adding a new virtual server, enter the basic settings, such as name and address.
4. In the Virtual Port section of the window, click **Create**. The **Create Virtual Port** window appears.
5. Enter a name for the virtual port, select the desired protocol (UDP or TCP), and enter the port number.

6. Click **General Fields** to display them.
7. Select Enabled next to **No Dest NAT**.
8. Click **Create**.
9. Click **Update** to complete the virtual server configuration.

CLI Configuration

Enter the `no-dest-nat` command at the configuration level for the virtual port:

IP NAT Support for VIPs

ACOS supports IP NAT for VIPs. This feature allows clients in a private network to connect to outside VIP servers, without revealing the client IP addresses to the servers. Dynamic NAT and static NAT are both supported.

The current release does not support this feature for FTP traffic.

Priority for Source IP NAT Configurations on Individual Virtual Ports

Source IP NAT can be configured on a virtual port in the following ways:

1. ACL-based source NAT (`access-list` command at virtual port level)
2. VIP source NAT (`slb snat-on-vip` command at global configuration level)
3. aFlex policy (`aflex` command at virtual port level)
4. Non-ACL source NAT (`source-nat` command at virtual port level)

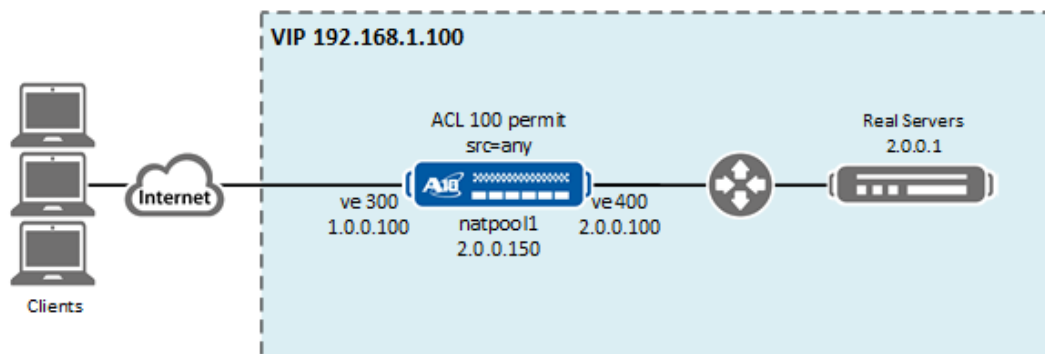
These methods are used in the order shown above for Layer 4 traffic. For Layer 7 traffic, VIP source NAT has priority over ACL-based source NAT.

For example (Layer 4 traffic), if IP source NAT is configured using an ACL on the virtual port, and the `slb snat-on-vip` command is also used, then a pool assigned by the ACL is used for traffic that is permitted by the ACL. For traffic that is not permitted by the ACL, VIP source NAT can be used instead.

VIP Source NAT

VIP Source NAT is an SLB feature that configures VIP ports to utilize global and interface SNAT data structures.

Figure 19 : Non-ACL NAT – NAT Pool and Smart NAT



To configure VIP Source NAT:

1. Configure a pool, range list, or static inside source NAT mapping that includes real IP addresses of the inside clients.
2. Enable inside NAT on the interface connected to the inside clients.
3. Enable outside NAT on the interface connected to the external VIP servers.
4. Enable SNAT on VIP – either globally or on individual ports.

The `snat-on-vip` command is available in global configuration level and in virtual port configuration level. When SNAT on VIP is globally configured, the feature is available on all ports. When SNAT on VIP is not globally enabled, it can be enabled on individual ports.

VIP Source NAT Configuration Example (CLI Example)

The following commands configure the elements that SNAT on VIP require.

- Configure NAT on the inside and outside interfaces:

```
interface ve 300
 ip address 1.0.0.100 255.255.255.0
 ip nat inside
```

```
interface ve 400
  ip address 2.0.0.100 255.255.255.0
ip nat outside
```

- Configure the access list and NAT pool.

```
access-list 1 permit any
ip nat pool natpool1 2.0.0.150 2.0.0.150 netmask /24
ip nat inside source list 1 pool natpool1
```

- Enable SNAT on VIP (globally).

```
slb common
  snat-on-vip
```

NOTE: When the traffic flows from client to server through the VIP, but the traffic is not evenly distributed across all the CPUs, NAT port allocation failure may occur before 64k per NAT IP is used. This is the expected behaviour of the NAT port allocation for VIPs. To resolve this issue, it is recommended to bind multiple NAT IP addresses instead of a single NAT IP address to the virtual port.

Using IP Pool Default Gateways

ACOS provides a feature to use the default gateway of an IP source NAT pool to forward traffic from a real server. When this option is enabled, the ACOS device checks the configured IP NAT pools for an IP address range that includes the server IP address (the source address of the traffic). If the address range in a pool does include the server's IP address, and a default gateway is defined for the pool, the ACOS device forwards the server traffic through the pool's default gateway.

CLI Configuration

- Use **snat-gwy-for-l3** option in transparent mode to forward layer 3 traffic using the default gateway of an IP source NAT pool:

```
ACOS(config)# slb common
ACOS(config-common)# snat-gwy-for-l3
```

- Use **snat-gwy-for-13 allow-in-gateway-mode** option in gateway mode to forward layer 3 traffic using the default gateway of an IP source NAT pool:

```
ACOS(config)# slb common
ACOS(config-common)# snat-gwy-for-13 allow-in-gateway-mode
```

Smart NAT for Virtual Ports

Smart NAT provides source NAT for virtual ports. The IP addresses that Smart NAT uses to create the mappings depend on whether VRRP-A high availability is enabled and floating-IP addresses are configured:

- With VRRP-A high availability – If VRRP-A high availability is configured, Smart NAT uses configured floating IP addresses as NAT addresses.
- Without VRRP-A high availability – If VRRP-A high availability is not configured, then Smart NAT uses IP address(es) on the ACOS interface connected to the real server.

In VRRP-A high availability deployments, if session synchronization is enabled, sessions created by Smart NAT are synchronized to the backup device.

- If you use VRRP-A high availability, it is recommended to bind a given service group to only a single virtual port. If you do bind a service group to multiple virtual ports, it is highly recommended to assign all the virtual servers to the same VRRP-A VRID.
- When a service group is bound to a virtual port, the Smart NAT resources are created for all the servers belonging to that service group. If the selected server does not have Smart NAT resources, then they are dynamically created. In this case, some initial connections may be dropped.
- Smart NAT applies only to ACOS devices deployed in route mode (also called “gateway” mode). The feature is not applicable to devices deployed in transparent mode.
- Smart NAT is not supported on SIP, SIP-TCP, or SIPS virtual ports.
- You can configure a virtual port to use both Smart NAT and a configured NAT pool. By default, configured pool addresses are used first and Smart NAT is used only when no more mappings are available in the configured pool.

- Optionally, you can configure Smart NAT to take precedence over the configured NAT pool. In this case, the configured pool is used only when there are no more available mappings using Smart NAT.
- If you do not use VRRP-A, 41K ports per interface IP address are used for Smart NAT mappings. ACOS can use the same ACOS interface IP address and port for more than one server connection. The combination of ACOS IP address and port number (source) and server IP address and port (destination) uniquely identifies each mapping.
- You can configure Smart NAT with IP-RR to round-robin across multiple floating IPs for VRRP-A or interface IPs without VRRP-A to reach more than 41K. In this case, the IP addresses rotates through the floating or interface IPs for every request to balance out the traffic across all the IPs.

Smart NAT uses all the multiple addresses if configured.

NOTE: All the service groups across the VIPs pointing to the same server must be on the same VRID. This means all the VIPs containing the service groups have to match.

GUI Configuration

Assuming you have an existing virtual server named vs1:

1. Navigate to the **ADC >> SLB >> Virtual Servers >> vs1 >> Virtual Port >> Create** page.
2. Expand the General Fields section.
3. Select the checkbox in the Source NAT Auto field.
4. If you want Smart NAT to be used before a pool is used.
5. Select the Precedence or IP-RR checkbox.
6. Click **Create** to save your changes.

CLI Configuration

The commands in this example configure two virtual ports. Smart NAT is enabled on each virtual port.

To begin, the following commands configure the data interfaces:

```
ACOS(config)# vlan 10
ACOS(config-vlan:10)# tagged ethernet 1
ACOS(config-vlan:10)# router-interface ve 10
ACOS(config-vlan:10)# exit
ACOS(config)# vlan 20
ACOS(config-vlan:20)# tagged ethernet 2
ACOS(config-vlan:20)# router-interface ve 20
ACOS(config-vlan:20)# exit
ACOS(config)# interface ethernet 3
ACOS(config-if:ethernet:3)# ip address 10.10.10.1 255.255.255.0
ACOS(config-if:ethernet:3)# exit
ACOS(config)# interface ve 10
ACOS(config-if:ve10)# ip address 172.31.10.1 255.255.255.0
ACOS(config-if:ve10)# exit
ACOS(config)# interface ve 20
ACOS(config-if:ve20)# ip address 192.168.10.1 255.255.255.0
ACOS(config-if:ve20)# exit
```

The following commands configure a source NAT pool, then return to the global configuration level:

```
ACOS(config)# ip nat pool snat-pool1 192.168.10.200 192.168.10.200 netmask /24
```

These commands configure real server “s1” with two TCP ports (80 and 21) and a service group for each port:

```
ACOS(config)# slb server s1 192.168.10.160
ACOS(config-real server)# port 80 tcp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# port 21 tcp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
ACOS(config)# slb service-group sg1-http tcp
ACOS(config-slb svc group)# member s1 80
ACOS(config-slb svc group-member:80)# exit
ACOS(config-slb svc group)# exit
ACOS(config)# slb service-group sg2-ftp tcp
ACOS(config-slb svc group)# member s1 21
ACOS(config-slb svc group-member:21)# exit
ACOS(config-slb svc group)# exit
```

The following commands configure the VIP. Smart NAT is enabled on each virtual port.

```
ACOS(config)# slb virtual-server vip1 192.168.10.150
ACOS(config-slb vserver)# port 80 tcp
ACOS(config-slb vserver-vport)# source-nat auto precedence
ACOS(config-slb vserver-vport)# source-nat pool snat-pool1
ACOS(config-slb vserver)# port 21 ftp
ACOS(config-slb vserver-vport)# source-nat auto
ACOS(config-slb vserver-vport)# source-nat pool snat-pool1
```

On the TCP virtual port, the **precedence** option is used with Smart NAT. This means Smart NAT is used first, then NAT pool is used only if all Smart NAT mappings are in use.

On the FTP virtual port, the **precedence** option is omitted. In this case, the source NAT pool is used first. Smart NAT is used only if no more mappings are available using the pool.

The following commands configure the VIP. Smart NAT with IP-RR is enabled on each virtual port.

```
ACOS(config)# interface ve 10
ACOS(config-if:ve:10)# ip address 10.211.1.1 255.255.255.0
ACOS(config-if:ve:10)# ip allow-promiscuous-vip
ACOS(config-if:ve:10)# ipv6 address FD00::BE2:54:34:2/7
ACOS(config-if:ve:10)# ipv6 enable
ACOS(config-if:ve:10)# exit
ACOS(config)# interface ve 20
ACOS(config-if:ve:20)# ip address 10.212.1.1 255.255.255.0
ACOS(config-if:ve:20)# ipv6 address FD00::BE2:54:35:2/7
ACOS(config-if:ve:20)# ipv6 enable
ACOS(config-if:ve:20)# exit
ACOS(config)# vrrp-a vrid 1
ACOS(config-vrid:1)# floating-ip 10.212.1.222
ACOS(config-vrid:1)# floating-ip 10.212.1.223
ACOS(config-vrid:1)# floating-ip 10.212.5.1
ACOS(config-vrid:1)# floating-ip 10.212.6.1
ACOS(config-vrid:1)# floating-ip FD00::BE2:54:35:131
ACOS(config-vrid:1)# floating-ip FD00::BE2:54:35:132
ACOS(config-vrid:1)# exit
ACOS(config)# slb server rs 10.212.1.2
```

```

ACOS(config-real server)# port 80 tcp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
ACOS(config)# slb server rs6 FD00::BE2:54:35:2
ACOS(config-real server)# port 80 tcp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
ACOS(config)# slb service-group sg-tcp tcp
ACOS(config-slb svc group)# member rs 80
ACOS(config-slb svc group-member:80)# exit
ACOS(config-slb svc group)# exit
ACOS(config)# slb service-group sgv6-tcp tcp
ACOS(config-slb svc group)# member rs6 80
ACOS(config-slb svc group-member:80)# exit
ACOS(config-slb svc group)# exit
ACOS(config)# slb virtual-server vs 10.211.1.131
ACOS(config-slb vserver)# port 8081 tcp
ACOS(config-slb vserver)# service-group sg-tcp
ACOS(config-slb vserver-vport)# source-nat auto ip-rr
ACOS(config-slb vserver-vport)# exit
ACOS(config-slb vserver)# exit
ACOS(config)# slb virtual-server vs6 FD00::BE2:54:35:6
ACOS(config-slb vserver)# port 8080 http
ACOS(config-slb vserver)# service-group sgv6-tcp
ACOS(config-slb vserver-vport)# source-nat auto ip-rr
ACOS(config-slb vserver-vport)# exit
ACOS(config-slb vserver)# exit

```

The following command shows Smart NAT statistics without ip-rr:

```

ACOS(config-slb vserver-vport)# show slb server s1 auto-nat-stats
Total Number of Services configured on Server s1: 23
Service      HA/VR ID   IP_RR   Nat Address   Port Usage   Total Used   Total
Freed   Failed
-----
s1:21/tcp   2           No      192.168.10.105   0           0
0           0
              192.168.10.104   0           0           0
              192.168.10.103   0           0           0
              192.168.10.106   0           0           0

```


192.168.10.107	0	0	0
192.168.10.108	0	0	0

The following command shows Smart NAT statistics with ip-rr:

```
ACOS(config-slb vserver-vport)# show slb server rs auto-nat-stats
Total Number of Services configured on Server rs: 21
Service      HA/VRID  IP_RR Nat Address      Port Usage Total Used   Total
Freed  Failed
-----
-----
rs:80/tcp    1          Yes 10.212.1.225    64      660469      660405
0
              10.212.1.226    64      660470      660406      0
rs:8080/tcp  1          Yes 10.212.1.225    0        0          0
0
              10.212.1.226    0        0          0      0
```

Virtual-port TCP Maximum Segment Life

You can customize the maximum Segment Life (MSL) for source-NAT connections virtual ports. The MSL is the maximum number of seconds a TCP segment (packet) is allowed to remain in the network. When one of the endpoints in a TCP connection sends a FIN to close the connection, that endpoint then enters the TIME-WAIT state.

During the TIME-WAIT state, the endpoint is not allowed to accept any new TCP connections. This behavior is meant to ensure that the TCP endpoint does not receive a segment belonging to a previous connection after the endpoint enters a new connection.

The TIME-WAIT state lasts up to twice the MSL. On some older TCP/IP stacks, this can result in a wait of up to 240 seconds (4 minutes) after a FIN before the endpoint can enter a new connection.

To reduce the time between connections for these endpoints, set the MSL for individual virtual ports, to 1-1800 seconds.

GUI Configuration

On the virtual port template configuration page, enter the desired value in the SNAT MSL field. Apply the template to the virtual port.

CLI Configuration

Use the **snat-msl** command to configure a custom MSL value for a virtual port. This example configures a source-NAT MSL of 18 seconds (Configuration commands do not include for the component service group):

```
ACOS(config)# ip nat pool natintf 192.168.20.48 192.168.20.48 netmask /24
ACOS(config)# slb template virtual-port ronvport
ACOS(config-vport)# snat-msl 18
ACOS(config-vport)# exit
ACOS(config)# slb virtual-server ronvip2 192.168.20.103
ACOS(config-slb vserver)# port 80 tcp
ACOS(config-slb vserver-vport)# service-group web
ACOS(config-slb vserver-vport)# source-nat pool natintf
ACOS(config-slb vserver-vport)# template virtual-port ronvport
```

Non-standard Port Support

ACOS provides full non-standard port support without using SNAT to preserve the original client IP address. This feature benefits in various scenarios, for example,

- Collect statistics and perform analysis based on client IP address.
- Use IP-address-based filters such as security groups or security group rules on Application Load Balancers to filter traffic.

To implement this feature, you can enable the 'port-translation-only' option in the SLB policy template under forward-policy actions: 'forward-to-internet', 'forward-to-service-group', and 'forward-to-proxy'. The forward-policy flow will select a unique internal NAT pool and only perform the source port translation.

This feature is supported in the following environments:

- Explicit proxy and SSLi/Dynamic Port SSLi
- Transparent proxy and SSLi/Dynamic Port SSLi

CLI Configuration

- To enable the feature, create an SLB policy template with forward-policy action and then bind it to inside virtual server/partition.

```
ACOS(config)# slb template policy p1
ACOS(config-policy)# forward-policy
ACOS(config-policy-forward-policy)# action aaa
ACOS(config-policy-forward-policy-action)# forward-to-internet http snat port-translation-only
ACOS(config-policy-forward-policy-action)# forward-to-service-group http snat port-translation-only
ACOS(config-policy-forward-policy-action)# forward-to-proxy http support-cert-fetch snat port-translation-only bypass
```

NOTE: The 'port-translation-only' command and 'NAT pool config' are mutually exclusive. When the 'port-translation-only' option is set, only source port translation will be done.

- To view the source port translation only NAT pool instance statistics in current partition, use the show command:

```
ACOS# show slb forward-policy-source-port-translation-only-stats
```

Group	Port	Usage	Total Used	Total Freed	Failed
IPv4	0	0	0	0	0
IPv6	0	0	0	0	0

GUI Configuration

- Navigate to **Security >> Forward Proxy >> Templates >> Create >> Policy**.
- Enable the **Port Translation Only** option.
- To view the source port translation statistics, navigate to **Security >> Forward Proxy >> Statistics >> Policy Source Port Translation Only** tab.

Dynamic Real Server Creation Using DNS

The following topics are covered:

Overview	137
Template Options	138
CLI Configuration	140
GUI Configuration	144

Overview

You can use DNS to simplify real server creation by specifying a DNS hostname instead of an IP address. In this case, the ACOS device periodically sends a DNS query for the hostname's IP address and dynamically creates a real server with the IP address returned by DNS. ACOS also creates a service-group member for the server, in each service group that contains the server.

Creating and Maintaining Dynamic Servers

To create and maintain dynamic real servers, the ACOS device sends a DNS query for each hostname real server at a configurable interval.

- If the DNS server replies with an Address (A) record for a hostname real server, the ACOS device creates the server or, if the server is already created, the ACOS device refreshes its TTL. ACOS also creates service-group members for the server and its ports.
- If the DNS server replies with a CNAME record, the ACOS device also sends a DNS query for the CNAME.

ACOS supports multiple servers with the same hostname. For example, if the DNS server replies with a different IP address for a hostname real server that has already been created, the ACOS device creates a second real server with the same hostname and the new IP address.

When the IP address returned by the DNS server matches the IP address of a statically configured real server, the server is not created.

Service groups can contain both static and hostname servers.

Dynamic Server Aging

Dynamically created real servers age out based on TTL values returned by the DNS server. ACOS sets a server's initial TTL when the server is created. The initial TTL value is the greater of:

- TTL value in the DNS reply
- DNS query interval multiplied by the min-ttl-ratio (described in [Template Options](#))

The server's TTL is decremented by 60 every minute. The TTL is refreshed each time the DNS server replies with the address. When the TTL reaches 0, the dynamically created server is removed. If the DNS server replies with the IP address after this, the server is dynamically created again.

- Dynamically created real servers have higher priority than statically created real servers, by default. If your configuration uses a combination of dynamically created real servers and statically created real servers, the dynamically created real servers are used more. This is true even if you leave the default load-balancing method, round robin, enabled. (To use round robin, see [GUI Configuration](#))
- When a dynamically created real server ages out, only that instance of the server (its port and service group member) is removed. Other instances (other IP addresses) for the same server (hostname) are not removed, unless they also age out. The real server configuration that you entered, used by the ACOS device to dynamically create servers, is not removed.

Template Options

The template options can be configured for static servers and ports also apply to dynamic servers and ports. In addition, server and server port templates have some new options, specifically for dynamic real servers.

These template options take effect when you apply a template to a dynamic server configuration. After this, any dynamic real servers that are created using the dynamic server configuration use the template values that were set when the template was applied to the dynamic server configuration, even if the values are later changed in the template.

Server Template Options

This topic describes the server template options for hostname real servers.

- `dynamic-server-prefix` – Specifies a short string to add to the front of the name for each dynamically created real server. Dynamically created servers are named using

the format: *prefix-ipaddr-hostname*

- The *prefix* is the string added by the ACOS device. You can specify a string of 1-3 characters. The default is “DRS”, for Dynamic Real Servers.
- The *ipaddr* is the IP address returned in the DNS reply.
- The *hostname* is the hostname you specify when you create the server configuration.

The maximum total length of a dynamic server name is 63 bytes. If the name becomes longer than 63 characters, the ACOS device truncates the name to 63 bytes.

- *dns-query-interval* – Specifies the interval which the ACOS device sends DNS queries for IP addresses of the dynamic real servers. You can specify 1-1440 minutes (one day). The default is 10 minutes.
- *max-dynamic-server* – Specifies maximum number of real servers that can be dynamically created for a specified hostname. You can specify 1-1023; default is 255. After the maximum number of servers is created, the oldest servers are deleted, determined by their creation time, to make room for new ones.
- *min-ttl-ratio* – Specifies the minimum initial value for the TTL of dynamic real servers. This option prevents dynamic real servers from aging out too quickly due to a small TTL value from the DNS server.
- To calculate minimum TTL value for a dynamic real server, the ACOS device multiplies *dns-query-interval* by *min-ttl-ratio*. For example, if *min-ttl-ratio* is 2 and *dns-query-interval* is 10 minutes (600 seconds), the minimum TTL for dynamic real servers is 1200. The *min-ttl-ratio* can be 1-15. The default is 2.

Server Port Template Options

This topic describes the server port template options for dynamic service-group members.

- *Dynamic-member-priority* and *decrement-delta* – Sets the initial priority of dynamic service-group members, and specifies how much to decrement from the priority after each DNS query.

Within a service group, member priority determine which members can service client requests. Normally, only the highest priority members can be used.

Decrementing priorities of dynamic members is a method to ensure the service group uses newer dynamically created members instead of older ones.

The initial priority can be 1-16, and the default is 16. The delta can be 0-8, and the default is 0.

Priority value decrements when the IP address is not refreshed after a DNS query. For example, assume a DNS query returns IP address 1.1.1.1, and the ACOS device creates a dynamic server with priority 16. However, the latest DNS query returns IP address 2.2.2.2 only. In this case, the priority of 1.1.1.1 is decremented by the delta value. If a later DNS query returns 1.1.1.1, the priority of server 1.1.1.1 is reset to 16.

If you leave the delta set to its default (0), service-group member priorities are not decremented.

Settings that apply to static servers and ports, such as connection and rate limits, apply individually to dynamically created servers or ports. For example, the connection-rate limit configured in a server template applies individually to each dynamically created server for a given hostname. The limit is not applied collectively to all dynamically created servers for the hostname.

CLI Configuration

This topic describes the CLI configuration example for dynamic real servers creation.

Configure Dynamic Real Server Creation

These commands configure hostname server parameters in a server port template and a server template:

```
ACOS(config)# slb template port temp-port
ACOS(config-rport)# dynamic-member-priority 12
ACOS(config-rport)# exit
ACOS(config)# slb template server temp-server
ACOS(config-rserver)# dns-query-interval 5
ACOS(config-rserver)# min-ttl-ratio 3
ACOS(config-rserver)# max-dynamic-server 16
ACOS(config-rserver)# exit
```


The following commands configure a hostname server, add a port to it, and bind the server template to it:

```
ACOS(config)# slb server s-test1 s1.test.com
ACOS(config-real server)# template server temp-server
ACOS(config-real server)# port 80 tcp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
```

The following commands configure a static real server:

```
ACOS(config)# slb server s-test2 10.4.2.1
ACOS(config-real server)# port 80 tcp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
```

The following commands configure a service group and add the hostname server and static server to it. The port template is bound to the member for the hostname server and port.

```
ACOS(config)# slb service-group sg-test tcp
ACOS(config-slb svc group)# member s-test1 80
ACOS(config-slb svc group-member:80)# template temp-port
ACOS(config-slb svc group-member:80)# exit
ACOS(config-slb svc group)# member s-test2 80
ACOS(config-slb svc group-member:80)# exit
ACOS(config-slb svc group)# exit
```

This command adds the DNS server to use for resolving the real server hostname into server IP addresses:

```
ACOS(config)# ip dns primary 10.10.10.10
```

The `show slb server detail` command displays detailed information for the hostname server. The command displays configuration details, followed by details for the dynamically created servers. See the Command Line Interface Reference for ADC for a description and examples of the command.

The following command displays service-group information. A separate row of information appears for each dynamically created member.

```
ACOS# show slb service-group
Total Number of Service Groups configured: 40
```

```

Current = Current Connections, Total = Total
Connections
Fwd-p = Forward packets, Rev-p = Reverse packets
Peak-c = Peak connections
Service Group Name
Service          Current Total      Fwd-p      Rev-p
Peak-c
-----
-----
*sg-test          State: All Up
DRS-10.4.2.6-s2.test.com:80    0          0          0          0          0
DRS-10.4.2.5-s1.test.com:80    36         1919        5714        5631
55
s-test2:80         0          53         265         212
311

```

The following command displays detailed statistics for the dynamically created service-group members:

```

ACOS# show slb service-group sg-test
Service group name: sg-test          State: All Up
Service selection fail drop:        0
Service selection fail reset:       0
Service peak connection:            0
Service: DRS-10.4.2.6-s2.test.com:80 UP
Forward packets: 0      Reverse packets: 0
Forward bytes: 0      Reverse bytes: 0
Current connections: 0      Persistent connections: 0
Current requests: 0      Total requests: 0
Total connections: 0      Response time: 0.00 msec
Total requests succ: 0
Peak conn: 0
Service: DRS-10.4.2.5-s1.test.com:80 UP
Forward packets: 5715      Reverse packets: 5631
Forward bytes: 546650      Reverse bytes: 919730
Current connections: 10      Persistent connections: 0
Current requests: 10      Total requests: 1919
Total connections: 1919      Response time: 0.00 msec
Total requests succ: 1877

```

```

Peak conn:          0
Service: s-test1:80          UP
Forward packets:    450      Reverse packets:    360
Forward bytes:      31500    Reverse bytes:    44820
Current connections: 0      Persistent connections: 0
Current requests:   0      Total requests:    0
Total connections:  90      Response time: 0.00 msec
Total requests succ: 1877
Peak conn:          0

```

The following command displays configuration information for the service group. In this example, the service group has dynamic members and a static member.

```

ACOS# show slb service-group sg-test config
Service group name: sg-test
Type: tcp                      Distribution: Round Robin
Health Check: None
Member Count:4
Member4: DRS-10.4.2.6-s2.test.com:80      Priority: 1
Member3: DRS-10.4.2.5-s1.test.com:80      Priority: 16
Member1: DRS-10.4.2.5-s-test2:80          Priority: 1
Member2: s-test1:80                       Priority: 1

```

Combining Dynamic and Static Real Servers

By default, dynamically created servers have the highest priority.

The following configuration contains a dynamically created server (s1) and a statically created server (s2). The default load-balancing method (round robin) is used, but s1 is used more than s2. To configure equal use of s1 and s2, the priority values for each server are explicitly set to the same value:

```

slb template port porttemp
  dynamic-member-priority 5
!
slb server s1 s1.com
  port 22 tcp
!
slb server s2 2.2.2.2
  port 22 tcp
!
slb service-group sg1 tcp

```

```
member s1 22
    priority 5
    template porttemp
member s2 22
!
slb virtual-server vs1 1.1.1.1
    port 22 tcp
    service-group sg1
```

Priority settings for dynamically created servers are set only using a port template, as shown in the example.

GUI Configuration

This topic described the GUI configuration of dynamic real servers creation.

Configure the server template

1. Navigate to **ADC >> Templates >> SLB**.
2. Click **New Template**. A drop-down menu appears.
3. Select **Server**. The Create Server Template dialog appears.
4. Enter a name for the template.
5. Configure these options. ([Template Options](#).)
 - a. DNS Query Interval
 - b. Dynamic Server Prefix
 - c. Min TTL Ratio
 - d. Max Dynamic Server
6. Click **Create**.

Create the real servers

1. Navigate to **ADC >> SLB >> Servers** tab.
2. Click **Create**.
3. In the **Name** field, enter a name for the real server.
4. Select the address **Type** radio button: IPv4, IPv6, or FQDN.

5. Enter the IP Address or FQDN hostname that is known to DNS.
6. Expand the Advanced Fields, and select the template from the **Template Server** drop-down menu.
7. Configure additional options for the real server and add ports, as applicable to your deployment.
8. When finished, click **Create**.

Configure a template for the server port

1. In the Port section of the page, click **Create**.
2. Enter number in the **Port Number** field.
3. Click **Advanced Fields** to expand and configure the advanced options for this server port.
4. To the far-right of Template Port, click the **Add+** link. The Create Port Template appears.
5. Enter a name for the port template in the **Name** field.
6. Configure these options. ([Template Options.](#))
 - a. Dynamic Member Priority
 - b. Decrement Value
7. Click **Create**.

Configure the service group

1. Navigate to **ADC >> SLB >> Service Groups** tab.
2. Click **Create**.
3. Enter a name for the service group in the **Name** field.
4. Click **Advanced Fields** to expand and configure the advanced options for this service group.
5. Select the Port template from the **Template Port** drop-down menu.
6. In the Member section, click **Create**.
7. Select the **Existing Server** radio button, and then select the **Server** from the drop-down menu.

8. Enter the number in the **Port** field.
9. Click **Create**.
10. Click **Update** to complete the service group configuration.

Virtual Port Ranges

The following topics are covered:

Overview	148
CLI Configuration	150
GUI Configuration	151

Overview

ACOS enables easier configuration for large ranges of virtual ports within a virtual-server configuration.

This feature may be helpful in deployments where it is desirable to have a VIP with a large number of different virtual ports. The ability to specify a port range under a single VIP simplifies network configuration, and it can be helpful within data centers or web-hosting companies that use port numbers to identify their specific customers.

Although similar performance can be achieved using wildcard VIPs, this approach does not scale well because wildcard VIPs limit the ACOS device to no more than 99 ACLs.

The virtual port range feature uses the range CLI option within a VIP configuration. A standard port number is specified within the VIP, and the range option is used to create an additional number of virtual ports, which will be added upon this base port.

For example, the base virtual port could be set to 80 and the range could be set to any value from 0-254. So if the range were set to the maximum of 254, then the virtual port range could start at 80 and go all the way up to 334 (80 + 254).

Virtual Port Types

The virtual port is internally identified as the port type and the base port. When using the show command to view statistics, you can only specify the base port. If you use the show command to try to view statistics for one of the auto-created virtual ports, a consolidated set of data and statistics will be provided for all virtual ports, starting from the base port and including all ports specified in the range.

Statistics and configurations are applied to the virtual port as a whole and not to the individual ports within the specified range.

This feature is supported on the following virtual port types:

- TCP
- HTTP

- TCP-proxy
- SSL-proxy
- HTTPS
- fast-HTTP

Preventing SLB Reset and Ethernet Statistics

ACOS provides an option to prevent resetting (clearing) of statistics for the following resources: SLB servers, service groups, virtual servers, and Ethernet interfaces.

By default, statistics counters for these types of resources can be reset by ACOS admins.

By default, clearing statistics is allowed. You can disable reset of SLB and Ethernet statistics, on a global basis.

This option applies to statistics counters in the output of the following:

GUI pages:

- **ADC >> Statistics >> L4**
- **ADC >> Statistics >> L7** and corresponding sub-pages
- **ADC >> Statistics >> Application** and corresponding sub-pages
- **ADC >> Statistics >> System** and corresponding sub-pages

CLI commands:

- `show slb virtual-server`
- `show slb service-group`
- `show slb server`
- `show interfaces`

Preventing SLB Reset and Ethernet Statistics

To disable reset of SLB and Ethernet statistics, use the `disable reset statistics` command at the global configuration level.

To enable statistics reset, use the **enable reset statistics** command at the global configuration level.

The following commands attempt to clear SLB and Ethernet statistics but are not allowed:

```
ACOS(config)# disable reset statistics
ACOS(config)# clear slb server all
Reset statistics disabled
ACOS(config)# clear statistics
Reset statistics disabled
```

CLI Configuration

This topic describes the CLI configuration example for adding virtual port to a virtual server.

The following commands create real servers “s1” and “s2”, binds them to a service group “sg1”, and then creates a VIP (“vip3”) at 40.40.40.4. TCP port 80 is created within the VIP configuration, and it has a range value of “9”, meaning that 9 virtual ports will be created from port 80-89, with port 80 as the base port. A secondary set of HTTP ports will be configured with a range of 5, meaning that the base port will be 90, and an additional five ports will be created from 91-95.

```
ACOS(config)# slb server s1 30.30.30.10
ACOS(config-real server)# port 80 tcp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
ACOS(config)# slb server s2 30.30.30.11
ACOS(config-real server)# port 80 tcp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
ACOS(config)# slb service-group sg1 tcp
ACOS(config-slb svc group)# member s1 80
ACOS(config-slb svc group-member:80)# exit
ACOS(config-slb svc group)# member s2 80
ACOS(config-slb svc group-member:80)# exit
ACOS(config-slb svc group)# exit
ACOS(config)# slb virtual-server vip3 40.40.40.4
ACOS(config-slb vserver)# port 80 tcp range 9
```

```
ACOS(config-slb vserver-vport) # service-group sgl
ACOS(config-slb vserver-vport) # template virtual-port vport1
ACOS(config-slb vserver-vport) # exit
ACOS(config-slb vserver) # port 90 http range 5
ACOS(config-slb vserver-vport) # service-group sgl
ACOS(config-slb vserver-vport) # template virtual-port vport1
ACOS(config-slb vserver-vport) # exit
ACOS(config-slb vserver) # exit
```

Note that the ports created with the **range** option must not overlap. If the TCP port range was set to 20 instead of 9 in the example above, then this would have caused a configuration error, because the TCP port range (80+20 = 100) would have overlapped with the HTTP port range (90-95).

In this example, if a client request were to come in at vip3, port 89, it would be sent to server port 80. However, if the port included in the destination address of the client request falls beyond the configured range (97, for example), then there would be no match and the packet would be discarded.

GUI Configuration

This topic describes the GUI configuration to add a virtual port range to a virtual server.

To specify a virtual port range within a VIP configuration:

1. Navigate to **ADC >> SLB >> Virtual Server**.
2. Click **+ Create**.
3. Enter a name for the virtual server.
4. Enter the base virtual port number in the Port field.
5. Enter the range (number of ports) in the Range field.
6. Click **Update** to complete the virtual server configuration.

Mapping Virtual IP Addresses and Real Ports

The following topics are covered:

Overview	153
CLI Configuration	157
GUI Configuration	158

Overview

ACOS supports the ability to auto-create mappings between a VIP and a real port on a real server. ACOS examines the IP address in a client's request, identifies the host portion, and then adds that number to the real port for a group of servers. In this way, the ACOS device can have many ports associated with a single VIP, and can deterministically control where incoming client requests are directed.

This feature is similar to [Virtual Port Ranges](#), in that it also leverages the `range` CLI command option. The `range` option allows you to specify the number of real ports that can be auto-mapped at the real server level. However, despite the use of this common CLI option, the two features are different.

While the “VIP to Real Port Mapping” feature creates a range of real ports on a real server and is used to map incoming requests to real ports on backend servers, the “Virtual Port Range” feature specifies a range of virtual ports within the VIP configuration and makes it faster and easier to configure large ranges of virtual ports within a virtual server configuration.

The following topics are covered:

Deterministic Mapping	153
Mapping VIP to Real Ports: Supported Virtual Port	155
Mapping VIP to Real Ports: Requirements	155
Virtual IP Creation Based on Subnet	156

Deterministic Mapping

ACOS can be configured as a subnet VIP, with “0” for the host portion of the address. For example, the VIP can be configured with an IP address such as 40.40.40.0 /24.

The value of the last octet configured as the ACOS device's VIP depends on the netmask length. The value can be “0”, but the following additional examples are equally valid:

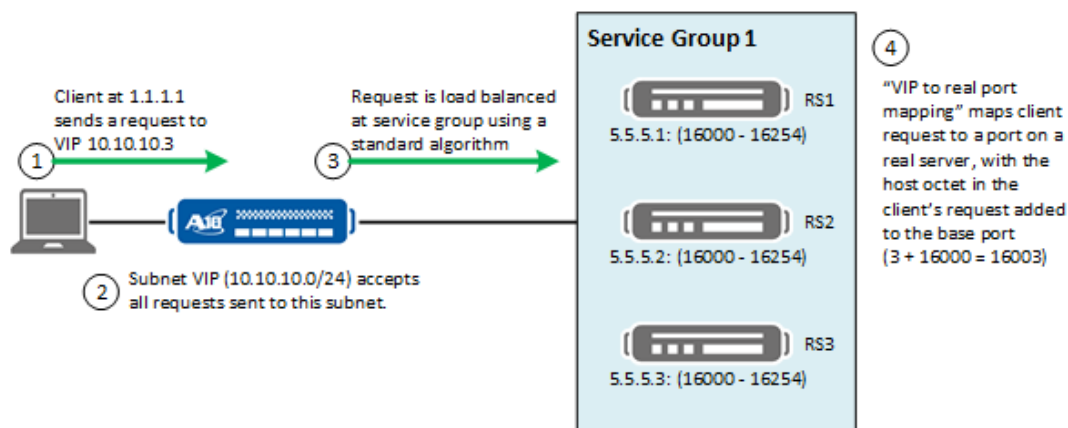
- 20.20.20.0 /24
- 20.20.20.240 /28
- 20.20.0.0 /16
- 20.20.20.252 /30

Configuring the ACOS device with a subnet VIP enables a single VIP to accept client requests from a large range of VIP addresses. Instead of requiring all client requests to go to 40.40.40.1, the host portion (last octet) can range from 0 – 254.

This feature creates a deterministic mapping between the host address in the client request and the real port on the backend servers. This mapping is achieved through a simple algorithm that adds the last octet in the destination VIP to the base port on the real server.

The host portion that appears in the client's request is added to the base port configured on the real servers. So, for example, if the client sends a packet to the VIP 10.10.10.3, then this last integer in the address ("3") is added to the base port configured on the backend servers (for example, 16000). The client's request will be mapped and forwarded to port "16000 + 3", or real port 16003. This is shown in [Figure 20](#).

Figure 20 : VIP to real port mapping



Example #1: A client request is sent to VIP 40.40.40.111 port 80, and it must be load balanced between three real servers having a port range from 16500–16550. (16500 is

the base port in this example.) Each one of the real servers in the service group has the same range of real ports.

ACOS adds the last octet of the VIP address (“111” for the VIP in this example) to the base port number on the real server (16500) to arrive at $16500 + 111$, or 16611.

Example #2: A client request is sent to the VIP at 216.69.188.4 port 80, and the packet must be load balanced between two real servers. Although each real server has a unique IP address, each server has the same range of ports. The base port is 16528 and the range is configured on the real server to be 254, so the range is from 16528–16782.

The last octet of the client’s destination address (“4” for this VIP) is added to the base port number on the real server ($16528 + 4$) to get a mapped real port of 16532.

Mapping VIP to Real Ports: Supported Virtual Port

This feature is supported on the following virtual port types:

- TCP
- HTTP
- HTTPS

Mapping VIP to Real Ports: Requirements

IPv6 is not supported

- The VIP’s prefix length must be less than 32.
- The host portion of the VIP address (last octet), can not be greater than the range value.
- If the client request has a large host portion (“100”), and the range configured on the real server is small (“5”), then there will be no mapping.

Virtual IP Creation Based on Subnet

ACOS provides a method to configure a range of virtual IP addresses (VIPs), based on IP subnet. Using this method, you can create a set of virtual servers that have contiguous IP addresses by specifying the beginning and ending IP addresses in the range.

The IP addresses in the specified range can not belong to an IP interface, real server, or other virtual server configured on the ACOS device.

The largest supported subnet length is /16.

Statistics are aggregated for all VIPs in the subnet virtual server.

In the GUI, only the first VIP in the range is visible.

GUI Configuration

1. Navigate to **ADC >> SLB >> Virtual Servers**.
2. Click **Create**.
3. In the **Name** field, enter a name for the virtual server.
4. Select the **Address Type** radio button: IPv4 or IPv6.
5. Enter the address in the **IP Address** field.

The IP address you specify here is the starting host address in the range and must be a valid host address. (For example, entering 192.168.1.0/24 is not valid.)

6. In the **Netmask** field, enter the Subnet mask. Do not use a space before or after the forward slash.
7. Click the **Advanced Fields**, and configure additional VIP options as required for your deployment.
8. When finished, click **Create**. The VIP appears in the VIP table.

CLI Configuration

The following command configures a set of VIPs for IP addresses 1.1.1.5-1.1.1.255:

```
ACOS(config) # slb virtual-server vs1 1.1.1.5 /24
```


If you do not specify a network mask, the virtual server is a standard VIP that has the IP address you specify as the *starting-ip* address.

CLI Configuration

This topic describes the CLI configuration example for VIP to Real Port Mapping.

Although similar to the “vport range” feature, the “VIP to real port mapping” feature configures the range option at the real server level, instead of at the VIP level. The “vport range” feature configures a range of virtual ports, whereas the “VIP to real port mapping” feature configures a range of real ports.

The following commands create real servers “s1” at 5.5.5.1 (with a real port range of 10), real server “s2” at 5.5.5.2 (with a range of 25), and real server “s3” at 5.5.5.3 (which does not have a range configured and will not be used for this feature).

Include the **range** option for each real server to be included in the service group, but only if you want that real server to be included in the mapping feature. The service group can be “mixed”. That is, some real servers within a service group can have the range option set, but it is not mandatory for all servers in a service group to be configured for “VIP to real port mapping”.

```
ACOS(config)# slb server s1 5.5.5.1
ACOS(config-real server)# port 80 tcp range 10
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
ACOS(config)# slb server s2 5.5.5.2
ACOS(config-real server)# port 80 tcp range 25
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
ACOS(config)# slb server s3 5.5.5.3
ACOS(config-real server)# port 80 tcp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
```

The following commands create service group “sg1” and bind the real servers to the service group:

```
ACOS(config)# slb service-group sg1 tcp
ACOS(config-slb svc group)# member s1 80
ACOS(config-slb svc group-member:80)# exit
```

```
ACOS(config-slb svc group)# member s2 80
ACOS(config-slb svc group-member:80)# exit
ACOS(config-slb svc group)# member s3 80
ACOS(config-slb svc group-member:80)# exit
ACOS(config-slb svc group)# exit
```

The **allow-vip-to-rport-map** command enables the VIP to Real Port Mapping feature for a subnet VIP. The virtual port template containing this option must be bound to the VIP, and the VIP itself must use a subnet for the last octet (e.g. 10.10.10.0 /24), or the feature will not work (Configuration commands for the vport1 virtual-port template are not included).

```
ACOS(config)# slb template virtual-port vport1
ACOS(config-vport)# allow-vip-to-rport-map
ACOS(config-vport)# exit
ACOS(config)# slb virtual-server vip3 10.10.10.0 /24
ACOS(config-slb vserver)# port 80 tcp
ACOS(config-slb vserver-vport)# service-group sg1
ACOS(config-slb vserver-vport)# template virtual-port vport1
ACOS(config-slb vserver-vport)# exit
ACOS(config-slb vserver)# port 90 http
ACOS(config-slb vserver-vport)# service-group sg1
ACOS(config-slb vserver-vport)# template virtual-port vport1
ACOS(config-slb vserver-vport)# exit
ACOS(config-slb vserver)# exit
```

GUI Configuration

This topic describes the GUI configuration of configuring VIP to real port mapping.

Setting the Port Range for a Real Server

1. Access the configuration page for the server ports:
 - a. Navigate to **ADC >> SLB >> Server** tab.
 - b. Click **Create**.
 - c. Enter the name and IP address of the server, in the Name and Host fields, respectively.
 - d. Click the **Create** button in the **Port** section.

2. Configure the port range:
 - a. Enter the base number for the range in the Port field.
 - b. In the Range field, enter the range of real ports you want to create within the real server configuration. This value can range from 0-254.
 - c. Click **Create**.
3. Click **Update** to complete the server configuration.

Adding the Real Servers to a Service Group

1. Access the configuration page for the service group:
 - a. Navigate to **ADC >> SLB >> Service Group** tab.
 - b. Click **Create**.
 - c. Enter a name for the service group.
 - d. Click the **Create** button in the Member section.
 - e. Select the server from the Server drop-down list.
 - f. Enter the base port number in the Port field.
 - g. Click **Create Member**.
2. Click **Update** to complete the service group configuration.

Enabling the VIP to Real Port Mapping within an SLB Virtual-port template

1. Access the configuration page for a virtual-port template:
 - a. Navigate to **ADC >> Templates >> SLB** tab.
 - b. Click **Create >> Virtual-Port**. The Create Virtual Port Template dialog appears.
 - c. Enter a name for the template.
 - d. Select the **Allow VIP To Real Port Mapping** check box.
 - e. Click **Create**.
2. Click **Update** to complete the service group configuration.

Binding the Service Group and Template to the VIP

The virtual port template containing this option must be bound to the VIP, and the VIP itself must use a subnet for the last octet (e.g. 10.10.10.0 /24), or the feature will not work.

1. Access the configuration page for the virtual service (virtual port):
 - a. Navigate to **ADC >> SLB >> Virtual Services** tab.
 - b. Click **Create**.
 - c. Enter a name for the virtual server. You can do either of the following:
 - i. Create a new virtual server using the Virtual Server Name field.
 - ii. Click **Use Existing Virtual Server** and enter the existing virtual server's name in the Server Name field.
 - d. Enter the virtual port number in the Port field.
 - e. Select the service group from the **Service Group** drop-down list.
2. Bind the virtual-port template to the virtual port.:
3. Click the **Bind link** under Templates.
4. Select **Virtual Port** from the drop-down list.
5. Select the template from the **Templates** drop-down list.
6. Click **Bind**.
7. Click **Update** to complete the virtual service configuration.

Part II: Application Delivery Controller Configuration

This section contains topics pertaining to application load balancing.

The following topics are covered:

- [MQTT Load Balancing](#)
- [FTP Load Balancing](#)
- [HTTP Load Balancing](#)
- [HTTP Options for SLB](#)
- [HTTP/2 Load Balancing](#)
- [HTTP Load Balancing to Proxy Servers](#)
- [Responding to Server Selection Failure](#)
- [SIP Load Balancing](#)
- [Financial Information eXchange Load Balancing](#)
- [Server and Port Templates](#)
- [Health Monitoring](#)
- [Alternate Real Servers and Ports for Backup](#)
- [Alternate Virtual Ports for Backup](#)
- [Priority Affinity](#)
- [Source-IP Persistence Override and Reselect](#)
- [Policy-Based \(PBSLB\) Templates](#)
- [Scan-All-Members Option \(Persistence\)](#)
- [Quality of Service Marking for TCP Traffic](#)
- [ADC Multi-PU Configuration](#)

MQTT Load Balancing

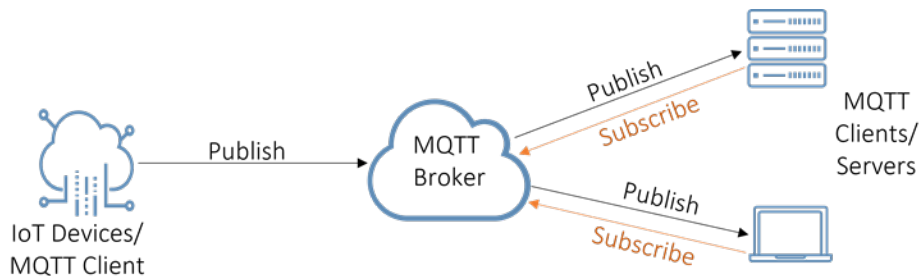
This section describes how to configure SLB for FTP services.

The following topics are covered:

Overview	162
Deployment Example	163
CLI Configuration	164

Overview

MQTT (Message Queuing Telemetry Transport) is an Oasis standard and an ISO/IEC PRF 20922 recommendation designed for Internet of Things (IoT) applications. It is a lightweight messaging protocol (Subscribe and Publish) that transports message from one device to another.



1. MQTT Client must be registered with the MQTT Broker indicating that it is available online to publish or subscribe to the message. A client can be a subscriber, a publisher, or both.
2. After the client is registered, it will send the list of topics that it intends to send (publish) and request for the list of topics that it wishes to receive (subscribe).

3. When topic/data needs to be distributed from the IoT devices or MQTT client, the publisher sends a control message with the data to the connected MQTT broker.
4. The MQTT broker publishes this information to MQTT clients that have subscribed to that topic.

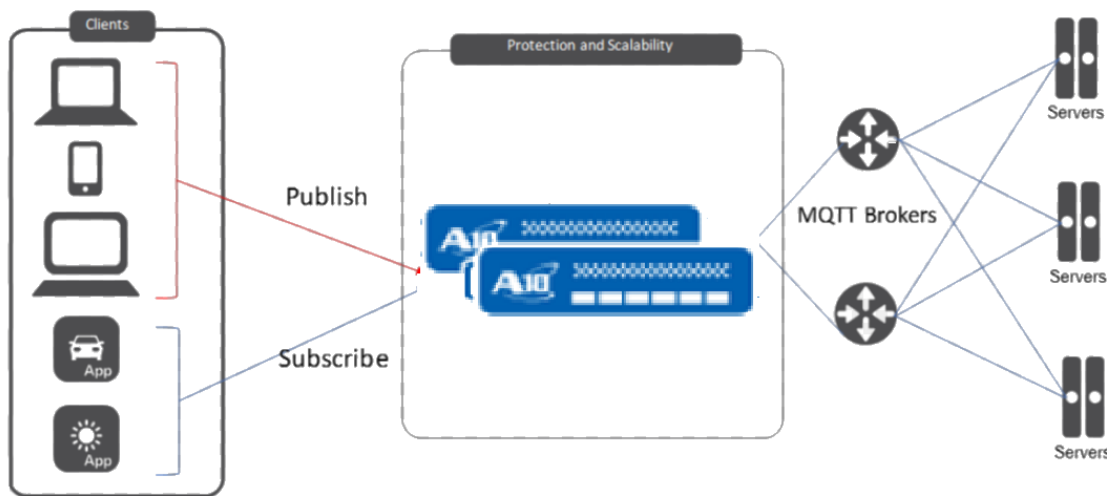
MQTT follows one-to-many message distribution and decoupling of applications unlike HTTP which defines request or response method. The protocol disregards the content of the payload and relies on TCP protocol for data transmission.

Deployment Example

When there are many subscribers for some data topics in the network, virtual port load balancing is critical to guarantee the service provided to these subscribers. By configuring an ACOS load balancer between the client device and a broker, you can analyze the type of content that is subscribed the most and which clients or users are subscribing to this content. You can bind aFlex on the virtual port to help collect this information.

The following image shows MQTT load balancing with ACOS. ACOS device receives the message from the client, parses the message and does the load balancing before sending it to the brokers.

Depending on the client registration, persistent connection is maintained with the broker by mapping the client ID that is sent during the registration and the server to which request is being forwarded.



CLI Configuration

- Configure an SLB template MQTT mq-1 and specify the hash-based algorithm for persistent client connection. Here the hashing is done based on the last 4 characters as of the client ID.

```
ACOS(config)# slb template mqtt mq-1
ACOS(config-mqtt)# clientid-hash-persist last 4
ACOS(config-mqtt)# exit
```

- The following commands configure a virtual server vs1 and virtual port 1883 for MQTT messages. Port 1883 is the default unencrypted port. Bind the SLB template to the virtual port.

```
ACOS(config)# slb virtual-server vs1 10.1.1.1
ACOS(config-slb vserver)# port 1883 mqtt
ACOS(config-slb vserver-vport)# template mqtt mq-1
```

- Use the following show command to see mqtt packet details.

```
ACOS# show slb mqtt
```

```
Total
```

```
-----
MQTT Connect          0
MQTT Connack          0
MQTT Publish          0
MQTT Puback           0
```


MQTT Pubrec	0
MQTT Pubrel	0
MQTT Pubcomp	0
MQTT Subscribe	0
MQTT Suback	0
MQTT Unsubscribe	0
MQTT Unsuback	0
MQTT Pingreq	0
MQTT Pingresp	0
MQTT Disconnect	0
MQTT Auth	0
MQTT Unknown	0
Current proxy conns	0
Total proxy conns	0
Total MQTT Commands	0
ACOS#	

FTP Load Balancing

This section describes how to configure SLB for FTP services.

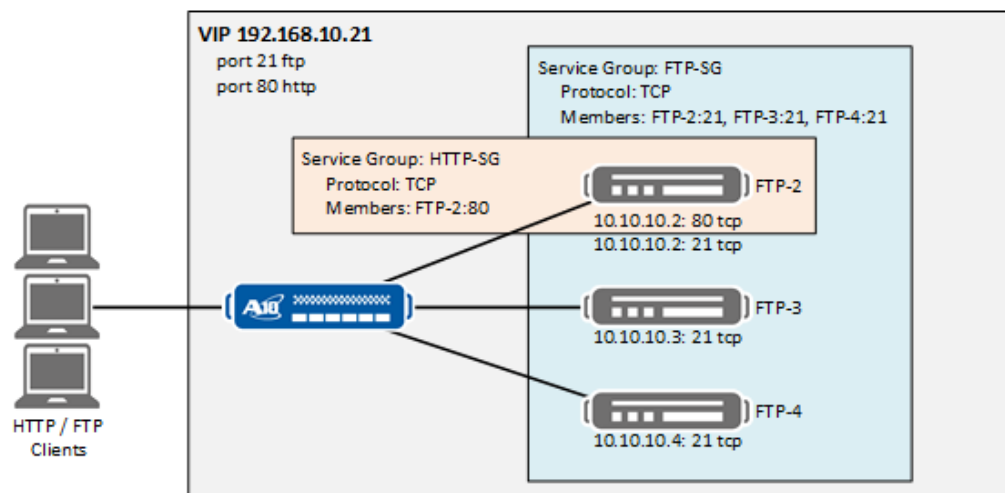
The following topics are covered:

Overview	167
CLI Configuration	168
GUI Configuration	168

Overview

FTP load balancing optimizes the download experience for clients by balancing FTP traffic across servers in a server farm. You can provide clients with a single, published virtual IP address for large files, and serve the files from a set of real servers. [Figure 21](#) shows an example of an FTP load balancing solution.

Figure 21 : FTP Load Balancing



In this example, FTP files are served by three real servers. Each server has the same set of files available for download. One of the servers also provides the HTML pages for the download site.

ACOS devices support both the passive and active (port) FTP modes.

The example uses the weighted-round-robin load balancing method. The ACOS device is configured to send all HTTP requests to server ftp-2. FTP requests will be load balanced among servers ftp-2, ftp-3, and ftp-4.

By assigning a weight to a real server and using a weighted load-balancing metric, you can bias load-balancing decisions to favor that server. This example assumes that the servers have equivalent capacity and performance. The differing weights compensate for the greater load to be placed on the server that is handling the HTTP requests.

CLI Configuration

The following commands configure the SLB server with FTP (port 21):

```
ACOS(config)# slb server rs
ACOS(config-slb server)# port 21 tcp
ACOS(config-slb vserver)# exit

ACOS(config)# slb service-group sg-ftp tcp
ACOS(config-slb service group)# member rs 21
ACOS(config-slb service group)# exit

ACOS(config)# slb virtual-server vs
ACOS(config-slb vserver)# port 21 ftp
ACOS(config-slb vserver-vport)# service-group sg-ftp
ACOS(config-slb vserver)# exit
```

GUI Configuration

This topic describes the GUI configuration example for FTP Load Balancing.

Configuring the health monitors

1. Navigate to **ADC >> Health Monitors**.
2. Click **Create**.
3. In the **Name** field, enter a name for the monitor.
4. Click **Method Type** drop-down menu and select **HTTP**.
5. Click **Create Monitor**. The new health monitor appears in the health monitor table.
6. Repeat step 2 through step 5 to configure the FTP health monitor. In step 4 select FTP instead of HTTP.

Configure the real servers

1. Navigate to **ADC >> SLB**.

2. Select the **Servers** tab from the menu bar.
3. Click **Create**.
4. Enter a name for the server in the **Name** field.
5. Select the **Type** radio button (IPv4, IPv6, FQDN).
6. Enter the IP address of the server in the **Host** field.

Configure the weight of the real servers

1. Still from within the Create Server page, click **Advanced Fields** to display additional configuration options.
2. In this example, change the value in the **Weight** field to 80.
3. In the Port section, click **Create**.
4. In the **Port Number** field, enter the number corresponding to HTTP (or FTP).
5. Leave the **Protocol** set to TCP.
6. In the Health Check radio button, click the **Monitor** radio button, and from the Health Monitor drop-down menu that appears, select the HTTP or FTP health monitor you just configured. (Select the appropriate health monitor that matches the port type you configured, either HTTP or FTP.)
7. Click **Create**. The new port appears in the port list.
8. Repeat the process to configure the real servers and the weight of the real servers for each of the other real servers you wish to add.

NOTE: The Health Monitor column shows the Layer 3 (ICMP ping) health monitors for the real servers, not the Layer4-7 health monitors for individual server ports. Click + to view the ports associated with the server.

To configure the TCP template for FTP

1. Navigate to **ADC >> Templates >> L4 Protocols** tab.
2. Click **Create** and select **TCP** from the drop-down menu.
3. Enter a name for the template in the Name field.
4. In the Idle Timeout field, enter 15000.
5. Click **OK**. The new template appears in the SLB template table.

Configure a service group for HTTP

1. Navigate to **ADC >> SLB >> Service Groups** tab.
2. Click **Create**.
3. Enter a name in the **Name** field.
4. Leave the transport **Protocol** set to TCP.
5. In the **Algorithm** field, select the load balancing method. For this example, select **Weighted Round Robin**.

Configure a service group member for HTTP

1. On the **Create Service Group** page, go to the **Member** section.
2. Click **Create**.
3. In the **Choose Creation Type** area, select the appropriate radio button (**Existing Server** or **New Server**).
 - If adding an existing server to this service group, select the Server drop-down and select the existing server.
 - If adding a new server to this service group, enter the real server's name and IP address.
4. Enter the port number for this protocol in the **Port** field.
5. Click **Create**. The server and port appear in the member list.
6. Click **Update** to update this service group. The new service group appears in the service group table.
7. Repeat this process of adding a service group and service group member for each combination of server and port. The following example shows adding member 10.10.10.2 for port 80 to service group http-grp.

Configure a service group for FTP

Repeat the procedure in [Configure a service group for HTTP](#), with the following differences:

- In the Algorithm drop-down list, select Weighted Round Robin. (If your configuration does not use weights to bias server selection, you can leave this

field set to Round Robin.)

- The following example shows adding members 10.10.10.2-4 for port 21.

Configure the virtual server

1. Navigate to **ADC >> SLB >> Virtual Servers** tab.
2. Click **Create**.
3. Enter a name in the **Name** field.
4. Enter the virtual IP address in the IP Address field. This is the IP address to which clients will send HTTP and FTP requests.

Configure the virtual server port

1. While still in the Create Virtual Server page, In the Virtual Port section, click **Create**. The Virtual Server Port section appears.
2. Enter a name in the **Name** field. This is optional.
3. In the **Protocol** drop-down list, select **service type**.

In this example, there are two services, HTTP and FTP. Select HTTP first and go to the next step

4. Edit the number in the **Port** field to match the protocol port that clients will request at the virtual IP address.
5. Select the service group that you configured in [Configure a service group for HTTP](#) from the **Service Group** drop-down list.
6. Click **Create**. The port and service group appear in the virtual port list.
7. Repeat this process to configure the virtual server port for the FTP service.

HTTP Load Balancing

This section describes HTTP load balancing and how to configure it.

The following topics are covered:

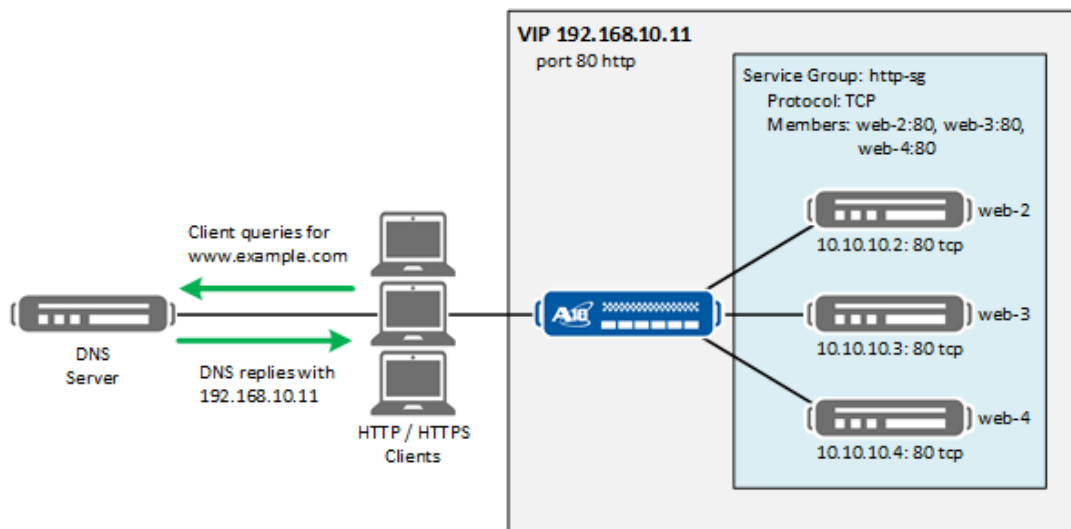
Overview	173
Components	174
Configuring an HTTP Template	178

Overview

HTTP load balancing manages HTTP traffic across a Web server farm. [Figure 22](#) displays an HTTP load balancing deployment.

Network topologies in application examples such as this one is simplified to focus on the application. For example, the Internet router connecting the clients to an ACOS device is not shown here. Likewise, a single ACOS device is shown. Your configuration might use an ACOS pair for VRRP-A high availability.

Figure 22 : HTTP Load Balancing



In this example, a server farm consisting of three servers provides content for Web site `www.example.com`. Clients access the site through its virtual IP address, `192.168.10.11`. When the ACOS device receives a client request for the HTTP port (80) on `192.168.10.11`, the ACOS device selects a real server and sends the client request to the server.

For simplicity in this example, the real servers use the default protocol port number for HTTP (80). The port numbers on the real and virtual servers are not required to match.

The client is unaware of the real IP address of the real server, nor is the client aware that the site actually consists of multiple servers. After selecting a real server, the ACOS device automatically performs the necessary Network Address Translation (NAT) (with source-nat configured, either source-nat auto or ip nat pools) to send the client request to the server, receive the reply from the server, and send the reply to the client. From the client's perspective, the Web session is between the client and port 80 on 192.168.10.11.

Fast-HTTP is optimized for high performance data transfer compared to regular HTTP. Due to this optimization, fast-HTTP does not support all comprehensive HTTP capabilities such as header insertion and manipulation. Do not use fast-HTTP for applications that require complete data transfer integrity.

Components

An HTTP load balancing configuration requires the following components:

- [Service Groups](#)
- [Virtual Server](#)
- [Templates](#)
- [Service Templates](#)
- [Server and Port Templates](#)
- [Health Monitors](#)

Service Groups

A service group contains a set of real servers from which the ACOS device can select to service a client request.

This example uses a single service group that contains all the real servers and the applicable service port (80). During configuration, you bind the service group to the virtual port(s) on the virtual server.

ACOS selects a server based on the load balancing method used by the service group, and on additional criteria relevant to the load balancing method.

In this example, the default load balancing method, round robin, is used. The round robin method selects servers in rotation. For example, the first client request is sent to server web-2, the next client request is sent to server web-3, and so on.

Virtual Server

The virtual server in this example has IP address 192.168.10.11 and virtual service port 80. When you configure a virtual service port, you specify the protocol port number for the port. You also specify the service type. ACOS supports the following service types for HTTP ports:

- HTTP / HTTPS – Complete TCP stack. Use this service type if you plan to customize any templates. For example, if you plan to use SSL (HTTPS load balancing or SSL offload), or customize the HTTP template to change information in the HTTP headers of server replies, use the HTTP service type. Also, use this service type for stream-based applications such as RAM Caching and compression.

The device supports HTTP/2 traffic, you can use `show session HTTP2` and `show slb HTTP2` commands to explicitly view HTTP/2 traffic sessions and statistics. For more information on HTTP/2, refer [HTTP/2 Load Balancing](#).

- Fast-HTTP – Streamlined hybrid stack for high performance. If you do not plan to offload SSL or customize any templates, use Fast-HTTP.

(For a complete list of the service types, see the **port** command in the *Command Line Reference*.)

Templates

Templates are sets of configuration parameters that apply to specific service types or to servers and service ports. This example uses the default settings for each of the templates that are automatically applied to the HTTP service type and to the real and virtual servers and ports. The rest of the information in this section is for reference but is not required reading to continue with this example.

For some templates types, the ACOS configuration has a “default” template that is automatically applied to a service port unless you apply another template of the same type instead.

Service Templates

For HTTP, the ACOS configuration applies “default” templates of each of the following template types to HTTP service ports:

- **TCP-Proxy** – TCP-proxy templates control TCP stack settings, including the idle timeout for TCP connections. Unless you need to change the setting for a TCP/IP stack parameter, you can safely allow the ACOS device to apply the default TCP-proxy template to the service types that use it. For more information, see 'slb template tcp-proxy' section in the *Command Line Reference*.
- **HTTP** – HTTP templates provide many options, including options to change information in the HTTP header, enable compression, and select a service group based on the URL requested by the client. By default, all the options in this template are disabled or not set, so you can safely allow the ACOS device to apply the default for this template type too. For more information, see 'slb template http' section in the *Command Line Reference*.
- **Connection Reuse** – Allows TCP connections between the ACOS device and real servers to be reused for multiple clients instead of terminating a connection and starting a new one for each new client. Although the default connection reuse template is automatically applied, the default settings in the template disable connection reuse. Unless you want to use connection reuse, you can ignore this template. (Connection reuse requires additional configuration. See [Connection Reuse](#).)

The following types of templates can also be used with HTTP service ports. However, these types of templates do not have “default” templates that are applied automatically.

- **Cookie Persistence** – Inserts a cookie in the HTTP header of a server reply before sending the reply to the client. The cookie ensures that subsequent requests from the client for the same virtual server and virtual port are directed to the same service group, real server, or real service port.
- **Source-IP Persistence** – Similar to cookie persistence, except the ACOS device does not insert cookies. Instead, clients are directed to the same resource in the server farm for every request, for the duration of a configurable timer on the ACOS device. The granularity of the persistence can be set to always use the same real server port, the same real server, or the same service group.

(For an example that uses a source-IP persistence template, see [Layer 4 TCP/UDP Load Balancing](#).)

Server and Port Templates

ACOS uses templates for configuration of some commonly used server and port parameters. By default, the following templates are applied:

- Default server template – Contains configuration parameters for real servers
- Default port template – Contains configuration parameters for real service ports
- Default virtual-server template – Contains configuration parameters for virtual servers
- Default virtual-port template – Contains configuration parameters for virtual service ports

Each of the default templates is named “default”.

For more information about server and port templates, see the following:

- [Server and Port Templates](#)
- “Config Commands: SLB Templates” section in the *CLI Reference*

Health Monitors

This example uses the following types of health monitors to check the real servers:

- Ping – A Layer 3 health method that sends an ICMP echo request to the real server’s IP address. The server passes the health check if the ACOS device receives a ping reply.
- TCP – By default, every 30 seconds the ACOS device sends a connection request (TCP SYN) to each load balanced TCP port on each server, in this case ports 80 and 443. A TCP port passes the health check if the server replies to the ACOS device by sending a TCP SYN ACK. By default, the ACOS device completes the TCP handshake.

In addition to these default health checks, you can configure health monitors for specific service types. This example uses an HTTP health monitor, with the following default settings.

- Every 30 seconds, the ACOS device sends an HTTP GET request for the default index page.
- The HTTP service port passes the health check if the requested page is present on the server and the server replies with an OK message (200).

For more information about health monitors and their configurable options, see [Health Monitoring](#).

Configuring an HTTP Template

To use the options in an HTTP template, you must configure the template, then bind the template to virtual service ports. You can bind an HTTP template to the following virtual service ports:

- HTTP
- Fast-HTTP
- HTTPS

To configure the HTTP load balancing solution described in [Overview](#), perform the following tasks on the ACOS device:

1. Configure an HTTP health monitor.
2. Configure the real servers. On each real server:
 - Add the HTTP service port.
 - Enable the HTTP health monitor.
3. Configure the service group. Add the real servers and service ports to the group.
4. Configure the virtual server:
 - Add the HTTP service port, with service type Fast-HTTP.
 - Bind the service group to the virtual port.

CLI Configuration

- To configure an HTTP template, enter the `slb template http` command at the global configuration level. To bind a template to a virtual service port, enter `template http` (port configuration level). These commands changes the CLI to the

template configuration level. The remaining sections in this section describe commands for configuring each option. The following example configures the HTTP template and binds the HTTP template to virtual port 80:

```
ACOS(config)# slb template http HTTP-TMP-1
ACOS(config-http)# exit
ACOS(config)# slb virtual-server vs1 1.1.1.1
ACOS(config-slb vserver)# port 80 http
ACOS(config-slb vserver-vport)# template http HTTP-TMP-1
```

- To decode HTTP body with the specified character set, use `default-charset` option. The following character sets are available:

- `iso-8859-1`
- `utf-8`
- `us-ascii`

The default character set is `utf-8`.

CLI Configuration

This topic describes the CLI configuration example for HTTP Load Balancing topology depicted in [HTTP Load Balancing](#).

Step 1: Configure the HTTP health monitor

```
ACOS(config)# health monitor http-monitor
ACOS(config-health:monitor)# method http
ACOS(config-health:monitor)# exit
```

Step 2: Configure the real servers

```
ACOS(config)# slb server web-2 10.10.10.2
ACOS(config-real server)# port 80 tcp
ACOS(config-real server-node port)# health-check http-monitor
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
ACOS(config)# slb server web-3 10.10.10.3
ACOS(config-real server)# port 80 tcp
ACOS(config-real server-node port)# health-check http-monitor
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
```

```
ACOS(config)# slb server web-4 10.10.10.4
ACOS(config-real server)# port 80 tcp
ACOS(config-real server-node port)# health-check http-monitor
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
```

Step 3: Configure the service group

```
ACOS(config)# slb service-group http-sg tcp
ACOS(config-slb svc group)# member web-2 80
ACOS(config-slb svc group-member:80)# exit
ACOS(config-slb svc group)# member web-3 80
ACOS(config-slb svc group-member:80)# exit
ACOS(config-slb svc group)# member web-4 80
ACOS(config-slb svc group-member:80)# exit
ACOS(config-slb svc group)# exit
```

Step 4: Configure the virtual server

```
ACOS(config)# slb virtual-server web-vip 192.168.10.11
ACOS(config-slb vserver)# port 80 fast-http
ACOS(config-slb vserver-vport)# service-group http-sg
ACOS(config-slb vserver-vport)# exit
ACOS(config-slb vserver)# exit
```

GUI Configuration

This section walks through individual GUI configuration pages. The GUI also provides smart templates for easy configuration. For information, see the GUI online help.

To configure an HTTP health method

1. Navigate to **ADC >> Health Monitors**.
2. Click **Create**.
3. In the **Name** field in the General Fields section, enter a name for the monitor.
4. Click the **Method Type** drop-down menu and select **HTTP** from the list.

The configuration fields change to those that apply to HTTP health monitors.

5. Optionally, select or enter additional options for the health monitor. (See [Health](#)

[Monitoring](#))

In this example, you can use the default settings.

6. Click **Create Monitor**. The new monitor appears in the health monitor table.

To configure the real servers

Perform the following procedure separately for each real server.

1. Navigate to **ADC >> SLB >> Servers** tab.
2. Click **Create**.
3. Enter a name for the server in the **Name** field.
4. Select the **Type** radio button (**IPv4**, **IPv6**, or **FQDN**), and then enter the address in the field below.

Enter the server's real address, not the virtual server IP address.

5. Click the **Health Monitor** drop-down menu and select a monitor, if one exists, or leave the monitor unset.

If you leave the monitor unset, the Layer 3 health monitor that comes in the ACOS configuration by default is used. (See [Default Health Checks](#))

To configure the real server port

1. While still within the Create Server window, in the Port section, click **Create**.
2. Enter the service port number on the real server in the **Port Number** field. In this example, enter "80".
3. For the **Health Check** radio button, select **Monitor**.
4. In the **Health Monitor** drop-down menu, select the HTTP health monitor configured in [To configure an HTTP health method](#).
5. Click **Create**. The port appears in the port list.

This example displays the server with the port created, followed by the list of all servers that were created.

ACOS begins sending health checks to a real server's IP address and service ports as soon as you finish configuring the server. The overall health status for the server is

shown in the Health column. If the status is Down, verify that health monitors are configured for all the service ports. The default Layer 3 health method is automatically used for the Layer 3 health check, unless you selected another health method instead.

To configure the service group

1. Navigate to **ADC >> SLB >> Service Groups** tab.
2. Click **Create**.
3. Enter a name for the service group in the **Name** field.
4. Select the load-balancing method from the **Algorithm** drop-down list.

For this example, you can leave the default selected: Round Robin

5. Click **Create** to create the service group. The new group appears in the service group table.

To configure the service group member

1. Click the **Edit** link for the service group you just created.
 2. In the Member section, click **Create**.
 3. Select the desired **Choose Creation Type** radio button:
 - Select **Existing Server**, and then click the **Server** drop-down and select the desired server, OR
 - Select **New Server**, and then enter the **Name**, address **Type**, and **Host** for the new real server.
 4. In the **Port** field, enter the service port number.
 5. Click **Create**.
 6. Click **Update**. The new group appears in the service group table.
- Repeat the process to configure the service group and service group members for each server and port.

To create the virtual server

1. Navigate to **ADC >> SLB >> Virtual Servers** tab.
2. Click **Create**. The Create Virtual Server window appears.
3. Enter a name for the virtual server in the **Name** field.
4. Select the **Address Type** radio button (**IPv4** or **IPv6**) and enter the address in the **IP Address** field. (This is the IP address that clients will request.)
5. Click **Create** to create the VIP. The new VIP appears in the list of virtual servers.

To configure the virtual server

1. Click **Edit** to add a virtual port to this VIP.
2. In the Virtual Port section, click **Create**. The Create Virtual Port section appears.
3. Enter a name in the **Name** field. Naming the virtual port is optional.
4. Click the **Protocol** drop-down menu and select the service type. In this example, select **Fast-HTTP**.
5. In the **Port** field, enter the service port number. In this example, enter "80".
6. Click the **Service Group** drop-down menu, and select the service group.
7. Click **Create**. The port appears in the Virtual Port list.

Viewing HTTP Statistics

ACOS provides real-time monitoring capabilities to various HTTP response codes for the HTTP, HTTPS, and HTTP ports of a virtual server. These new statistics communicate whether 5xx HTTP response codes originated from the real server or ACOS. An immediate display for the event cause allows you to quickly and effectively respond to HTTP events.

NOTE: In the current release, HTTP response codes are not written to event logs.

Viewing HTTP Status Code Statistics (CLI Example)

Enter the `show slb http-proxy` command to display counters for HTTP response codes. See the “Command Line Interface Reference for ADC” for information about the command.

For example:

```
ACOS-Active(config) # show slb http-proxy vs800-http 80
```

	Total
status code 1XX	3
status code 2XX	1
status code 3XX	12
status code 4XX	8
status code 5XX	2
status code 6XX	3
.	
.	
.	
Rsp time < 200m	0
Rsp time < 500m	1
Rsp time < 1s	3
Rsp time < 2s	7
Rsp time < 5s	13
Rsp time >= 5s	22

HTTP Options for SLB

HTTP templates provide various SLB options. Some options control real servers or service groups selection, while other options modify HTTP header information or enhance website performance.

Similarly, Fast-HTTP is optimized for high performance information transfer compared to regular HTTP. Due to this optimization, fast-HTTP does not support all comprehensive HTTP capabilities such as header insertion and manipulation.

NOTE: It is recommended not to use fast-HTTP for applications requiring data transfer integrity.

The following topics are covered:

Server and Service Group Selection Options	186
Performance Enhancing Options	210
HTTP Requests Options	222
Server Replies Options	231
HTTP Protocol Compliance	233

Server and Service Group Selection Options

HTTP template provides the following options to select real servers or service groups as per the business needs. These options takes precedence over the load-balancing selection method. By default, ACOS uses the load-balancing method set for the service group to select a real server.

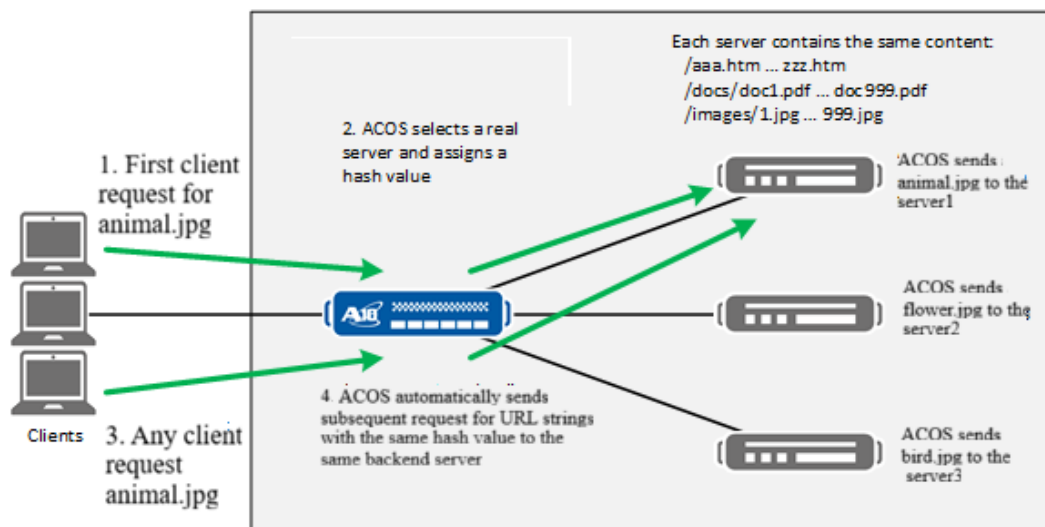
- URL hash switching – Selects a real server based on a hash value calculated from part of the URL string. ([URL Hash Switching](#))
- URL / host switching – Selects a *service group* based on the URL path or domain in the client's GET request. ([URL / Host Switching](#))
- URL Failover – If the URL in GET request cannot be reached due to server unavailability, the ACOS device sends a 302 Redirect to the client. ([URL Failover](#))
- 5xx retry and reassignment – Retries a server that replies to a request with a 5xx status code instead of sending the status code to the client, and reassigns the request to another server if the first server continues to reply with a 5xx status code. ([5xx Retry and Reassignment](#))
- Strict transaction switching – Performs server selection for each request within a client-server session, rather than performing server-selection once per session. This option provides a simple method to force rebalancing of server selection. ([Strict Transaction Switching](#))
- Non-HTTP bypass – Redirects non-HTTP traffic to a specific service group. This feature helps prevent non-HTTP traffic from being dropped by the ACOS device. ([Non-HTTP Bypass.](#))

URL Hash Switching

URL hash swapping is a technique for optimizing a server farm in which different servers deliver the same content. This feature improves website performance by caching content on the real servers.

When enabled, URL hashing chooses a real server for the initial request for a specific content and assigns a hash value to the URL. ACOS sends successive content requests to the same real server.

Figure 23 : URL Hashing



[Figure 23](#) shows an example of URL hashing and the workflow is explained below:

- A service group contains three real servers. Each of the real servers contains the same set of animal.jpg, flower.jpg, and bird.jpg files.
- ACOS is configured to calculate a hash value based on the URL string in the client request, and assign the hash value to that URL.
- ACOS device sends all the request with the same URL, which has the same hash value to the same backend server.
- If the real server becomes unavailable, the ACOS device selects another server, assigns a hash value to it, and uses that server for all subsequent requests for URL strings that have the same hash value.

Hash Options

The following options are available in URL hash switching:

- Offset – Specifies how far into the string to begin hash calculation.
- Bytes – Specifies how many bytes of the URL string to use to calculate the hash value.

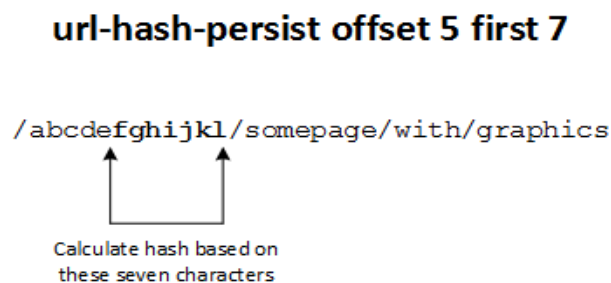
- First or last – Specifies which end of the URL string to use to calculate the hash value.
- Server load awareness – Allows servers to act as backups to other servers, based on load.

Offset

You can specify an offset at which to begin when calculating a hash value. By default, there is no offset.

[Figure 24](#) displays an example of calculating a hash value starting with the fifth character in the URL string.

Figure 24 : URL Hashing Offset Example



Server Load Awareness

Server load awareness allows servers to act as backups to other servers, based on their load. Normally, URL hashing selects a server for the first request for a given URL, then uses the same server for all subsequent requests for the same URL. In cases where a given URL becomes wildly popular (for example, a viral video), the server for that URL can become overwhelmed.

The server load awareness option provides a way to avoid server outage in such situation. After some configuration on the server and on the ACOS device, the ACOS device can learn a server's load status from the server.

Configuring Hash Switching with Server Load Awareness

Configuring Hash Switching with Server Load Awareness (Example)

Here is an example of how server load awareness works. In this example, URL hash switching is used to balance traffic load across three servers: S1, S2, and S3.

Assume that the first request for URI /article-new1 is hashed to S1. Subsequent requests are load balanced as listed in [Table 3](#).

Table 3 : Server-Status Load-Balancing Example

Server	Load Status Reported by Server	ACOS Action
S1	0	New requests for /article-new1 are sent only to server S1.
S2	0	
S3	0	
S1	2	New requests for /article-new1 are distributed between S1 and S2, using round robin.
S2	0	
S3	0	
S1	2	New requests for /article-new1 are distributed between S1 and S3, using round robin.
S2	1 or 2	
S3	0	

Configuring Hash Switching with Server Load Awareness (Server Procedure)

This feature requires custom configuration on the server. The server must be configured to insert an HTTP header named “Server-Status” in the server’s responses. The header must have one of these values: 0, 1, or 2.

```
Server-Status: load=N
```

The value of *N* can be 0, 1, or 2.

ACOS manages requests to the server based on the Server-Status code. [Table 4](#) describes the valid load status codes that can be reported by a server.

Table 4 : Server-Status Load Value

Load Status Reported by Server	Description	ACOS Action
0	Server is able to handle all of its own requests. Server also can handle requests for other servers if necessary.	ACOS device continues using the server for the URLs hashed to the server. If necessary, ACOS device also uses the server to help with URLs hashed to servers that have load status 2.
1	Server is able to handle its own requests. However, server can not handle requests for other servers.	ACOS device continues using the server for the URLs hashed to the server. ACOS device does not use the server to help handle requests for other servers.
2	Server is overloaded and needs help handling its own requests. Requests are distributed among this server and at least one other server (with load status 0), in round robin fashion.	ACOS device uses servers that have load status 0 to help handle the overloaded server's requests. If no other servers are able to help, all servers in the service group are pulled in to help. Requests will be sent round-robin among the busy servers. For example, if there are 3 servers, and s1 returns status 2, s2 returns 1, and s3 returns 0, then the traffic is sent round-robin between s1 and s3. However, if s3 returns 1 or 2, then the traffic is sent round-robin among all 3 servers.

The system conditions that result in reporting 0, 1, or 2 depend on how you program calculation of the code. For example, you can program the server to set the Server-Status code based on CPU utilization, memory utilization, I/O utilization, and so on.

For a CPU-intensive application, you could calculate the Server-Status code based on CPU utilization. For an I/O-intensive application, you could calculate the Server-Status code based on I/O utilization.

Configuring Hash Switching with Server Load Awareness (ACOS Configuration)

On the ACOS device, URL hash switching with server load awareness does not require configuration of dedicated backup servers in the service group. Instead, any primary server can also act as a backup for other servers, based on server load.

Server load awareness is disabled by default but can easily be enabled in new or existing URL hash switching configurations. Configure an HTTP template with URL hash switching. Include the use-server-status (CLI) or Use Server Status (GUI) option.

Configuring URL Hashing

The following sections show how to configure URL hashing using both CLI and GUI:

CLI Configuration

The following commands implement the URL hashing configuration shown in [URL Hashing](#). The configuration of the sg1 service group referenced in the example is not included.

```
ACOS(config)# slb template http hash
ACOS(config-http)# url-hash-persist last 6
ACOS(config-http)# url-switching ends-with .htm service-group sg1
ACOS(config-http)# exit
ACOS(config)# slb virtual-server vs1 1.1.1.1
ACOS(config-slb vserver)# port 80 http
ACOS(config-slb vserver-vport)# service-group sg1
ACOS(config-slb vserver-vport)# template http hash
```

The following commands configure an HTTP template for URL hash switching with server load awareness:

```
ACOS(config)# slb template http url-hash
ACOS(config-http)# url-hash-persist last 12 use-server-status
```

The following commands configure an HTTP template to perform URL hashing with the offset shown in [URL Hashing Offset Example](#):

```
ACOS(config)# slb template http url_hash1
```

```
ACOS(config-http) # url-hash-persist offset 5 first 7
```

GUI Configuration

1. Access the configuration sections for the template. (See [Configuring an HTTP Template](#).)
2. By default, the URL Hash Persist radio button is set to Disabled. To activate the configuration fields, select one of the following radio buttons:
 - First – creates hash using first *N* characters of the URL. You must specify *N* in **URL Hash First** field.
 - Last – creates hash using last *N* characters of the URL. You must specify *N* in **URL Hash Last** field.
 - Offset – Enables Offset option. You must specify the number of characters in **URL Hash Offset** field.
3. If you plan to use the server load awareness option, select the **Enable** radio button for Use Server Status.
4. Click **Create**.

GUI Configuration

1. Access the configuration sections for the template. (See [Configuring an HTTP Template](#).)
2. By default, the URL Hash Persist radio button is set to Disabled. To activate the configuration fields, select one of the following radio buttons:
 - First – creates hash using first *N* characters of the URL. You must specify *N* in **URL Hash First** field.
 - Last – creates hash using last *N* characters of the URL. You must specify *N* in **URL Hash Last** field.
 - Offset – Enables Offset option. You must specify the number of characters in **URL Hash Offset** field.
3. If you plan to use the server load awareness option, select the **Enable** radio button for Use Server Status.
4. Click **Create**.

URL / Host Switching

ACOS supports multiple service groups. URL / host switching enables an ACOS device to select a service group based on the URL or domain name in a client's GET request. The selection overrides the service group configured on the virtual port.

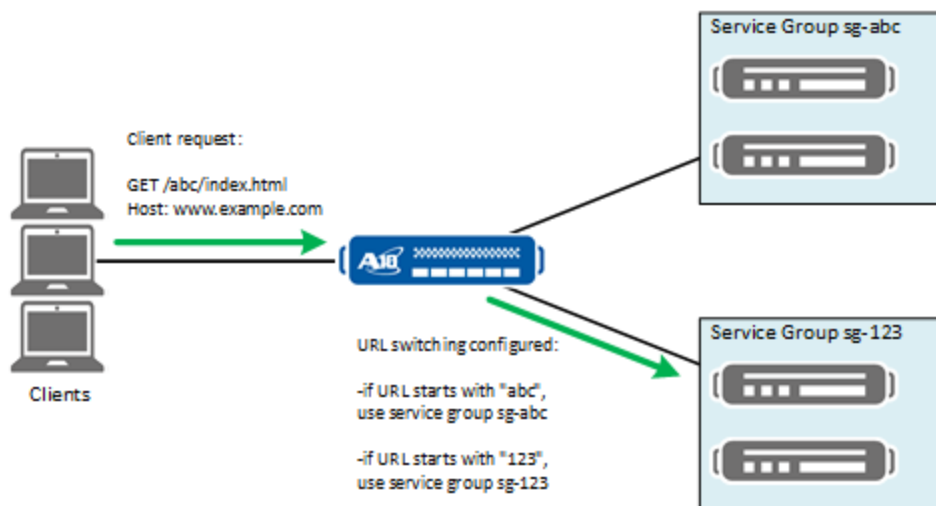
You can configure an HTTP template with one of the following service-group switching options:

- URL switching – Selects a service group based on the URL path in the GET line of the HTTP request's header.
- Host switching – Selects a service group based on the domain name in the Host field of the HTTP request's header

If you plan to use URL / host switching along with cookie persistence, you must enable the match-type service-group option in the cookie persistence template. (See [Using URL / Host Switching with Cookie Persistence](#).)

[Figure 25](#) shows an example of URL switching.

Figure 25 : URL Switching



provide content for `www.example.com/abc`. The real servers in service group `sg-123` provide content for `www.example.com/123`.

URL switching rules configured on the ACOS device select a service group based on the beginning of the URL on the GET line of client requests. Requests for URLs that begin with `"/abc"` are sent to service group `sg-abc`. Likewise, requests for URLs that begin with `"/123"` are sent to service group `sg-123`.

An HTTP template can be configured with only one type of service-group switching, URL switching *or* host switching. However, if you need to use both types of switching, you can do so with an aFlex script.

Match Options

URL / host switching selects a service group based on rules that map part of the URL string or host (domain name) to the service group. You can use the following match options in URL / host switching rules:

- Starts-with *string* – matches only if the URL or host name starts with the specified string.
- Contains *string* – matches if the specified string appears anywhere within the URL or host name.
- Ends-with *string* – matches only if the URL or host name ends with the specified string.

These match options are always applied in the following order, regardless of the order in which the rules appear in the configuration. The service group for the first match is used.

- Equals
- Starts-with
- Contains
- Ends-with

If a template has more than one rule with the same option (starts-with, contains, or ends-with) and a URL or host name matches on more than one of them, the most-specific match is used. For example, if a template has the following rules, requests for host `"www.ddeeff.org"` will always be directed to service group `http-sgf`:

```
host-switching contains d service-group http-sgd
```

```
host-switching contains dd service-group http-sge
host-switching contains dde service-group http-sgf
```

If you use the starts-with option with URL switching, use a slash in front of the URL string. For example:

```
url-switching starts-with /urlexample service-group http-sg1
```

Configuring URL / Host Switching

The following sections show how to configure URL / host switching using both GUI and CLI.

CLI Configuration

The commands to implement the URL switching configuration shown in [URL Switching](#) is performed in the below steps:

1. Configure the HTTP template:

```
ACOS(config)# slb template http urlswitch
ACOS(config-http)# url-switching starts-with /abc service-group sg-abc
ACOS(config-http)# url-switching starts-with /123 service-group sg-123
ACOS(config-http)# exit
```

2. Configure the Virtual Server:

In the example, the added `urlswitch http template` will forward the flows as per the configuration. Anything that is not matching with the template is redirected to the `service-group sg-abc`, which has been configured.

```
ACOS(config)# slb virtual-server vs1 1.1.1.1
ACOS(config-slb vserver)# port 80 http
ACOS(config-slb vserver-vport)# template http urlswitch
ACOS(config-slb vserver-vport)# service-group sg-abc
```

In the example, the added `urlswitch http template` will forward the flows as per configuration. Anything that is not matching with the template, the device returns 504 response.

```
ACOS(config)# slb virtual-server vs1 1.1.1.1
ACOS(config-slb vserver)# port 80 http
ACOS(config-slb vserver-vport)# template http urlswitch
```

GUI Configuration

1. Access the configuration sections for the template. (See [Configuring an HTTP Template](#).)

The configuration fields for Host Switching and URL Switching is within the HTTP template. You can configure either option (but not both) since they are mutually exclusive.

2. Select Host Switching:
 - a. Click **Add**.
 - b. Click the **Switching Type** drop-down list and select Starts With, Ends With, Contains, Equals or Host Hits Enable.
 - c. In the **Match String** field, enter the required data to match upon.
 - d. Click the **Service Group** drop-down list, select the service group to which to send client requests.
3. Select URL Switching:
 - a. Click **Add**.
 - b. Click the **Switching Type** drop-down list and select Starts With, Ends With, Contains, Equals, URL Case insensitive, or Enables URL Hits.
 - c. In the **Match String** field, enter the URL to search upon.
 - d. In the **Service Group** drop-down list, select the service group to which to send client requests.
4. Click **OK**. The HTTP template list appears.
5. To save the configuration, click **Save** on the right top of the window.

GUI Configuration

1. Access the configuration sections for the template. (See [Configuring an HTTP Template](#).)
2. The configuration fields for Host Switching and URL Switching are present at the bottom of the template window. You can configure either option (but not both), since they are mutually exclusive.
3. For Host Switching:

- a. Click **Switching Type** drop-down; select Starts With, Ends With, Contains, Equals or Host Hits Enable.
 - b. In the **Match String** field, enter the host name to search upon.
 - c. Click the **Service Group** drop-down list, select the service group to which to send client requests.
 - d. Click Add.
4. For URL Switching:
 - a. Click the **Switching Type** drop-down and select Starts With, Ends With, Contains, Equals, URL Case insensitive, or Enables URL Hits.
 - b. In the **Match String** field, enter the URL to search upon.
 - c. In the **Service Group** drop-down list, select the service group to which to send client requests.
 - d. Click Add.
5. Click **OK**. The HTTP template list appears.

Using URL / Host Switching with Cookie Persistence

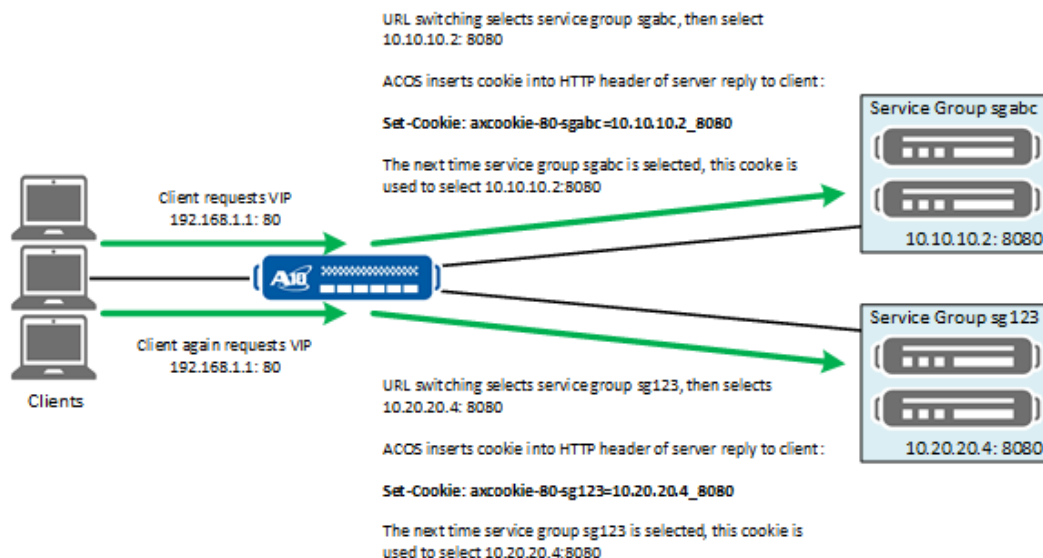
ACOS supports URL / host switching with cookie persistence in the same SLB configuration. To enable this support, you must enable the match-type service-group option in the cookie persistence template.

By default, the service-group option is disabled in cookie persistence templates. In this case, URL switching or host switching is used only for the initial request from a client. After the initial request, the same service group is always used for subsequent requests from the same client.

To continue using URL or host switching to select a service group for each request, enable service-group option in the cookie persistence template. For each request from the client, the ACOS device first selects a service group, then uses information in the cookie to select the real server and port within the service group.

[Figure 26](#) shows an example.

Figure 26 : URL Switching with Cookie Persistence



In this example, URL switching and cookie persistence are both configured, and the service-group option is enabled in the cookie persistence template. For each client request, URL switching selects a service group first. Then, after a service group is selected, a real server and port are selected within the service group.

- If the client's request does not have a persistence cookie that includes the selected service group, the ACOS device uses SLB to select a server, then inserts a persistence cookie into the reply from the server. The cookie includes the service group name.
- If the client's request already has a persistence cookie containing the name of the selected service group, the device uses the information in the cookie to select the same server in the service group.

For example, the first time service group sgabc is selected by URL switching, ACOS inserts a cookie into the server's reply to ensure the same server is used when URL switching again selects sgabc. The first time service group sg123 is selected by URL switching, the device inserts a second cookie into the server's reply to ensure the same server is used when URL switching selects sg123. Even though URL switching does not always select the same service group, the same server within the selected service group is always selected.

Cookie Persistence Match-Type Options

When cookie persistence is configured, the ACOS device adds a persistence cookie to the server reply before sending the reply to the client. The client's browser re-inserts the cookie into each request. The format of the cookie depends on the match-type setting:

- **match-type (port)** – This is the default setting. Subsequent requests from client are sent to the same real port on the same real server. URL switching or host switching is used only for the first request.

The cookie that the ACOS device inserts into the server reply has the following format:

Set-Cookie: *cookieName-vport=rserverIP_rport*

The *vport* is the virtual port number. The *rserverIP* is the real server IP address and the *rport* is the real server port number.

The port option is shown in parentheses because the CLI does not have a “port” keyword. If you do not set the match type to server (see below), the match type is automatically “port”.

- **match-type server** – Subsequent requests from the client for the same VIP will be sent to the same real server, provided that all virtual ports of the VIP use the same cookie persistence template with match-type set to server. URL switching or host switching is used only for the first request.

The cookie that the ACOS device inserts into the server reply has the following format:

Set-Cookie: *cookieName=rserverIP*

- **match-type (port) service-group** – Subsequent requests from the client will be sent to the same real port on the same real server, within the service group selected by URL switching or host switching. URL switching or host switching is still used for every request.

The cookie that the ACOS device inserts into the server reply has the following format:

Set-Cookie: *cookieName-vport-servicegroupName=rserverIP_rport*

- match-type server service-group – Subsequent requests from the client for the same VIP will be sent to the same real server, within the service group selected by URL switching or host switching. URL switching or host switching is still used for every request.

The cookie that the ACOS device inserts into the server reply has the following format:

Set-Cookie: *cookieName-servicegroupName=rserverIP*

For security, address information in the persistence cookies is encrypted.

Cookie Persistence Match-Type Option in Pass-Thru Mode

ACOS can also forward the cookie in pass-thru mode, so no information is modified, and the set-cookie header is not parsed from the server packet response or from subsequent client responses. In this case, the server is the one that sends the persist cookie. If the Set-Cookie header contains additional attributes, then the ACOS configuration should match those. Any mismatch between the ACOS configuration and the server persist cookie will break the persistence behavior.

To use pass-thru mode, there needs to be an `httpd.conf` file on the server. In the CLI, you can generate the pass-thru cookie that needs to be added to the server's configuration file. This cookie is generated based on match type and parameters that you specify.

Pass-thru mode cannot be enabled in conjunction with standard ACOS persist cookie configurations.

Using URL / Host Switching with Source IP Persistence

By default, if URL / host switching is configured along with source IP persistence, the URL / host switching settings are not used. Instead, the default service group is always selected. To enable URL / host switching to be used along with source IP persistence, you must use the match-type service-group option in the source IP persistence template.

For more information, see the description of the `slb template persist source-ip` command in the “Config Commands: SLB Templates” section of the Command Line Interface Reference.

CLI Configuration

The following commands configure a cookie persistence template named “persist-cookie-sg” and enable port-level persistence with support for URL switching or host switching:

```
ACOS(config)# slb template persist cookie persist-cookie-sg
ACOS(config-cookie persist)# name SGCookie
ACOS(config-cookie persist)# match-type service-group
ACOS(config-cookie persist)# exit
ACOS(config)# exit
```

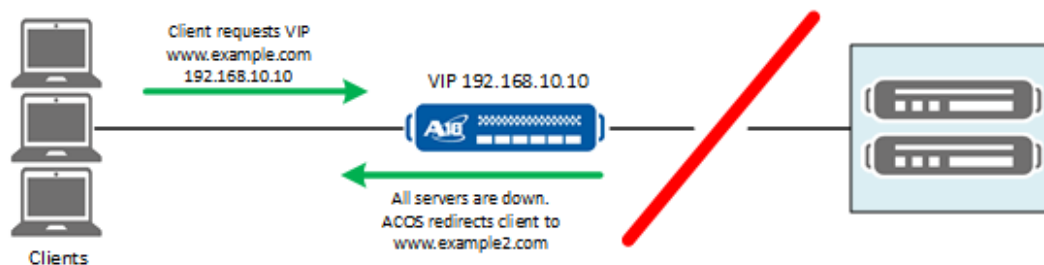
The following commands create the SLB persist cookie template and names it “passive-persist.” This temple enables pass-thru mode for passive cookie persistence.

```
ACOS# gen-server-persist-cookie passive-persist match-type service-group
sg1 32 64 192.168.100.100
Server Persist Cookie-Value (edit Expires/Path): "passive-persist-8192-
sg1=GEGEKIMAEAAA;Expires=Fri, 15-May-2026 22:50:04 GMT;Path=/"
ACOS# config
ACOS(config)# slb template persist cookie passive-persist
ACOS(config-cookie persist)# pass-thru
ACOS(config-cookie persist)# exit
```

URL Failover

ACOS can send an HTTP 302 Redirect message to a client when the real servers for the URL requested by the client are unavailable. [Figure 27](#) shows an example.

Figure 27 : URL Failover



In this example, a client sends a request for `www.example.com` (virtual IP address `192.168.10.10`). However, this VIP is unavailable because all the real servers are failing their health checks. ACOS is configured to send an HTTP 302 Redirect message if the VIP is down, redirecting clients to `www.example2.com`.

By default, URL failover is not configured. To configure it, you specify the URL to which to redirect clients. Like the other HTTP options, you can apply this option to a virtual port by configuring the option in an HTTP template, and binding the template to the virtual port.

The URL failover option does not affect redirect messages sent by real servers. To alter redirect messages from real servers, use the URL redirect-rewrite option instead. (See [URL Redirect Rewrite](#))

Configuring URL Failover

The following sections show how to configure URL failover.

GUI Configuration

1. Access the configuration sections for the template. (See [Configuring an HTTP Template](#).)
2. In the **Failover URL** field (at the top of the HTTP Template page), enter the URL where clients are to be redirected.
3. Click **OK**. The template is listed with other HTTP templates.

CLI Configuration

The following commands implement the URL failover configuration shown in [URL Failover](#).

The following commands configure the HTTP template:

```
ACOS(config)# slb template http urlfailover
ACOS(config-HTTP template)# failover-url www.example2.com
ACOS(config-HTTP template)# exit
```

The following commands bind the HTTP template to virtual port 80:

```
ACOS(config)# slb virtual-server vs1 1.1.1.1
ACOS(config-slb vserver)# port 80 http
```

```
ACOS(config-slb vserver-vport) # template http urlfailover  
ACOS(config-slb vserver-vport) # exit  
ACOS(config-slb vserver) # exit
```

Decompression for HTML Tag Insertion

For Over-the-top (OTT) services, proxy server was deployed inline to insert HTML codes in HTTP payload in server responses., ACOS devices support this feature with aFlex for non-compressed traffic. This feature is extended to support HTML tag insertion for HTTP compressed traffic (response) in TCS.

Running with aFlex, the compressed traffic is decompressed and an HTML tag is added to the response. The response is then recompressed and sent back to the client.

As an example, on receiving the responses of HTTP traffic, the load balancer can insert a reference to JavaScript. If the traffic is compressed, then ACOS device decompresses the traffic, adds a reference to JavaScript, recompresses the response, and sends it back to the client.

Configuring Content Compression

The following sections show how to configure content compression.

GUI Configuration

Navigate to the configuration page to create a new HTTP template:

1. Navigate to **ADC >> Templates >> L7 Protocols** tab.
2. Click **+ Create** and select **HTTP** from the drop-down menu.
3. Enter a name for the HTTP template.
4. Scroll to Compression and click to expand and display a set of additional fields.
 - **Auto Disable On High CPU** – Specify a value from 1-100 percent to configure an automatic disable of HTTP compression based on CPU utilization.
 - **Keep Accept Encoding** – This option configures ACOS to leave the Accept-Encoding header in HTTP requests from clients instead of removing the header. When this option is enabled, compression is performed by the real server instead of the ACOS device when the server is configured to perform

compression. The ACOS device compresses content that the real server does not compress.

- **Level** – Click the drop-down menu and select a value ranging from 1-9. Each level provides a higher compression ratio, beginning with level 1, which provides the lowest compression ratio. A higher compression ratio results in a smaller file size after compression. However, higher compression levels also require more CPU processing than lower compression levels, so performance can be affected.
 - **Minimum Content Length** – Specify the minimum number of bytes the content must be in order to be eligible for compression. The range is 1- 2147483647 and the default is 120.
5. To add more content types to be compressed:
 - a. In the Compression Content Type field, enter the string for a content type to compress.
 - b. Click **Add**.
 - c. Repeat step a and step b for each type of content to compress.
 6. Click **OK**.
 7. If your ACOS device supports hardware-based compression, enable the feature:
 - a. Select **ADC >> SLB >> Global**.
 - b. Click the checkbox next to **Hardware Compression**.
 - c. Click **Update**.

Hardware Compression is not available when the ACOS device does not contain a compression module.

CLI Configuration

These commands configure an HTTP template (http-compress) that uses compression level 5 for files with media type "application" or "image". Files of media type "application/zip" are excluded from compression.

```
ACOS(config)# slb template http http-compress
ACOS(config-http)# compression enable
ACOS(config-http)# compression level 5
ACOS(config-http)# compression content-type image
```



```
ACOS(config-http) # compression exclude-content-type application/zip
ACOS(config-http) # exit
```

This command displays HTTP compression statistics. The counters are in bytes and apply to all HTTP compression configured in all HTTP templates on the ACOS device.

```
ACOS(config) # show slb http-proxy
```

```
Total
```

```
-----
Curr Proxy Conns 58
Total Proxy Conns 49
HTTP requests 306
HTTP requests(succ) 269
No proxy error 0
Client RST 17
Server RST 0
No tuple error 0
Parse req fail 0
Server selection fail 0
Fwd req fail 0
Fwd req data fail 0
Req retransmit 0
Req pkt out-of-order 0
Server reselection 0
Server premature close 0
Server conn made 50
Source NAT failure 0
Tot data before compress 1373117
Tot data after compress 404410
```

The following commands enable hardware-based compression and display statistics for the feature:

```
ACOS(config) # slb common
ACOS(config-common) # hw-compression
ACOS(config-common) # exit
ACOS(config) # show slb hw-compression
Hardware compression device is installed.
Hardware compression module is enabled.
Total
-----
total request count      177157
```

```
total submit count      177157
total response count    177157
total failure count     0
last failure code       0
compression queue full  0
max queued request count 84
max queued submit count 68
```

Temporary Compression Disable During High CPU Utilization

ACOS can temporarily disable HTTP compression when CPU utilization reaches a specified threshold. When utilization returns below the threshold, ACOS re-enables HTTP compression. This feature can help overall system performance by temporarily freeing CPU resources used for compression to perform other tasks.

This option is disabled by default. You can enable it in individual HTTP templates.

- The feature operates on individual CPUs. Example: Utilization on CPUs 1 and 3 is over the threshold and below the threshold on other CPUs, HTTP compression is disabled only on CPUs 1 and 3.
- This feature applies to software-based compression. If hardware-based compression is enabled, this feature is inapplicable and has no effect.

GUI Configuration

Navigate to the configuration page to create an HTTP template:

1. Navigate to **ADC >> Templates >> L7 Protocols** tab.
2. Click **+ Create** and select **HTTP** from the drop-down menu.
3. Enter a name for the HTTP template.
4. Scroll to Compression and click to expand and display a set of additional fields.
5. In the **Auto Disable On High CPU** field, specify a value from 1-100 percent to configure an automatic disable of HTTP compression based on CPU utilization.
6. Click **OK**.

CLI Configuration

To configure automatic disable of HTTP compression based on CPU utilization, use the `compression auto-disable-on-high-cpu` command at the configuration level for the HTTP template. The following commands disable HTTP compression when CPU utilization is greater than 50%.

```
ACOS(config)# slb template http HTTP-TMP-1
ACOS(config-http)# compression auto-disable-on-high-cpu 50
ACOS(config-http)# exit
```

To display statistics for software-based compression, use the `show slb compression` command.

Strict Transaction Switching

By default, the ACOS device performs server selection once for a client session. After the initial selection, all requests within that session are automatically sent to the same real server. A new server is selected during the session only if the server that was originally selected is no longer available.

If the load among real servers appears to be unbalanced, you can enable strict transaction switching to rebalance the load. The strict transaction switching option forces the ACOS device to perform server selection for each request within every session.

Use this option only if needed, and disable the option once the server load is rebalanced. This option makes server selection much more granular but also uses more ACOS system resources.

Enabling Strict Transaction Switching

The following sections show how to enable strict transaction switching.

GUI Configuration

1. Navigate to **ADC >> Templates >> L7 Protocols** tab.
2. Click **+ Create** and select **HTTP** from the drop-down menu.
3. Enter a name for the HTTP template.

4. Click the Strict Transaction Switch checkbox.
5. Click **Create**. The HTTP template appears in the SLB template list.

CLI Configuration

To enable strict transaction switching, enter the `strict-transaction-switch` command at the configuration level for the HTTP template.

5xx Retry and Reassignment

By default, if a real server replies to a request with a 5xx status code (such as, HTTP 503 – Service Unavailable), the ACOS device forwards the error to the client. HTTP templates can override this behavior. by configuring the ACOS device to retry sending a client's request to a service port that replies with an HTTP 5xx status code, and reassign the request to another server if the first server replies with a 5xx status code. ACOS can reassign the request up to the configured number of retries.

For example, assume a service group has three members (s1, s2, s3), and the retry is set to 1. Then, if s1 replies with a 5xx status code, the ACOS device reassigns the request to s2. If s2 also responds with a 5xx status code, the ACOS device cannot reassign the request to s3, because the maximum number of retries has been used.

Depending on the configured 5xx retry option, either the service port and server remain eligible for more client requests, or the ACOS device stops sending client requests to the service port and server for 30 seconds.

- Server re-selection is not performed if Layer 3 features such as PBSLB, source-IP persistence, or cookie persistence are configured on the virtual port. These features override the server re-selection.
- Use of this HTTP template option also requires the `strict-transaction-switch` option to be used in the same HTTP template. (See [Strict Transaction Switching](#).)
- This option is supported only for virtual port types HTTP and HTTPS. It is not supported for fast-HTTP or any other virtual port type.

Configuring 5XX Retry and Reassignment (CLI Example)

These commands configure an HTTP template to reselect a server when the initially selected server responds 4 times to a client's request with a 5xx code. ACOS stops

using the service port and server for 30 seconds after reassignment.

```
ACOS(config)# slb template http 5xxretry
ACOS(config-http)# strict-transaction-switch
ACOS(config-http)# retry-on-5xx
```

Non-HTTP Bypass

Non-HTTP bypass redirects non-HTTP traffic to a specific service group. This feature helps prevent non-HTTP traffic from being dropped by the ACOS device.

GUI Configuration

1. Navigate to **ADC >> Templates >> L7 Protocols** tab.
2. Click **+ Create** and select **HTTP** from the drop-down menu.
3. Enter a name for the template.
4. Click the **Bypass Non-HTTP traffic** checkbox and choose the server group in the Bypass Service Group field. This is the server to which non-HTTP traffic will be redirected.
5. Select or enter values for the template options you want to use. The remaining sections in this section describe the fields for configuring each option.
6. Click **OK**. The template appears in the ADC template list.

CLI Configuration

To redirect non-HTTP traffic to a specific service group use the `non-http-bypass service-group` command (HTTP template configuration level). By default, ACOS drops non-HTTP requests sent to an HTTP port:

To configure, view, or remove this feature, follow these steps:

1. On an ACOS, configure the HTTP template and indicate the service group (in this case `sg1`) to which all non-HTTP traffic should be sent:

```
ACOS(config)# slb template http http
ACOS(config-http)# non-http-bypass service-group sg1
```

2. To view the existing configuration, use the `show slb template` command:

```
ACOS(config-http)# show slb template | section sg1
```

```
slb service-group sg1 tcp
  priority 10 drop
  priority 5 reset
  non-http-bypass service-group sg1
```

3. Bind the HTTP template to the virtual port 80:

```
ACOS(config)# slb virtual-server vip1 10.10.10.66
ACOS(config-slb vserver)# port 80 http
ACOS(config-slb vserver-vport)# name _10.10.10.66_HTTP_80
ACOS(config-slb vserver-vport)# service-group sg1
ACOS(config-slb vserver-vport)# template http http
ACOS(config-slb vserver-vport)# exit
ACOS(config-slb vserver)# exit
```

The above commands help apply the template to the virtual-port. When the http template is applied to the “virtual port 80,” any non-http requests will be forwarded to the service-group specified in the non-http bypass option.

4. To remove the non-http-bypass option for the service group (`sg1`), issue the following command:

```
ACOS(config)# slb template http http
ACOS(config-http)# no non-http-bypass service-group sg1
```

Performance Enhancing Options

- HTTP Packet Flow Modes – ACOS devices define two HTTP/HTTPS proxy packet flow modes that specify device method of managing multi-request data transfers. ([HTTP Packet Flow Modes](#))
- High-speed HTTP Content Replacement – Allows quick configuration of content replacement in HTTP replies from load-balanced servers. ([High-speed HTTP Content Replacement](#))
- Content Compression – Configures the device to offload content compression from real servers. Enabling content compression on the ACOS device increases website

performance by freeing real server resources from CPU-intensive compression tasks. ([Content Compression](#))

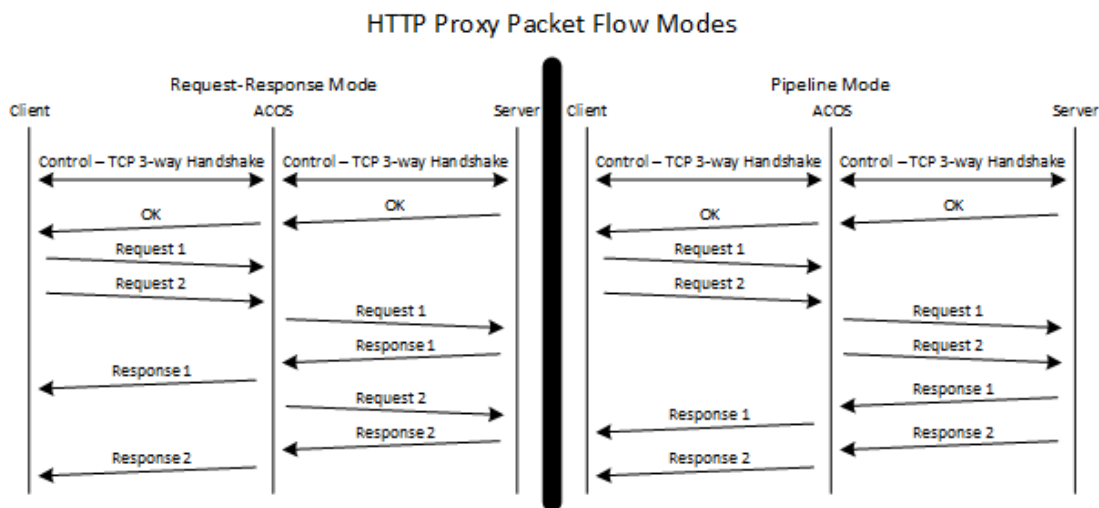
- File Inspection Service – Detects malware on files in HTTP/2 data streams through an ICAP based Cylance server that inspects files using artificial intelligence and machine based learning algorithms. File Inspection is described in the *SSL Insight Configuration Guide*.

HTTP Packet Flow Modes

ACOS devices define two HTTP/HTTPS proxy packet flow modes: Request-response and pipeline. These modes specifies the device method of managing multi-request data transfers. [Figure 28](#) displays data flow diagrams for the two modes.

- Request-response mode: The ACOS device does not send subsequent requests until it receives a response to previous requests.
- Pipeline mode: The ACOS device can send multiple requests that define a data transfer to the same server without waiting for responses to previous requests.

Figure 28 : HTTP and HTTPS Data Packet Flow Modes



Pipeline is the default HTTP and HTTPS packet flow mode. There are no commands that explicitly specify the mode that a device uses. The only method to manually specify a packet mode is to configure a condition that enables Request-Response mode.

Request-response mode is enabled when the device is performing per-request load balancing. Enabling any of the following features also enables request-response mode and subsequently disables pipeline mode.

- Strict translation switch (implemented through an SLB HTTP template)
- URL host persist (implemented through an SLB HTTP template)
- Compression is enabled (implemented through an SLB HTTP template)
- Failover is enabled (implemented through an SLB HTTP template)
- Redirect response is enabled (implemented through an SLB HTTP template)
- Non-HTTP bypass is enabled (implemented through an SLB HTTP template)
- Retry is enabled (implemented through an SLB HTTP template)
- Connection reuse is enabled (implemented through an SLB connection-reuse template)
- Persistence is configured on virtual port (implemented through an SLB persist template)
- RAM cache is enabled on virtual port (implemented through an SLB cache template)
- External service is enabled on virtual port (implemented through an SLB external-service template)
- Flex includes retry

High-speed HTTP Content Replacement

ACOS provides an HTTP template option for quick configuration of content replacement in HTTP replies from load-balanced servers. When the ACOS device detects the specified data pattern in a server reply, the ACOS device replaces the matching content with the content you specify.

A maximum of 8 content-replacement rules are supported in a given HTTP template. If you need more rules, aFlex supports up to 32.

HTTP Content Format

ACOS uses either a Content-Length header or a Transfer-Encoding: chunked header for the new content.

- If the replacement pattern is the same length as the original pattern, ACOS uses the same header type used by the server.
- If the replacement pattern length is different from the length of the original pattern, the ACOS device uses a Transfer-Encoding: chunked header and chunk-encodes the content.

Configuring High-Speed HTTP Content Replacement

GUI Configuration

Navigate to the configuration page to create a new HTTP template:

1. Navigate to **ADC >> Templates >> L7 Protocols** tab.
2. Click **+ Create** and select **HTTP** from the drop-down menu.
3. Enter a name for the HTTP template.
4. Scroll down and click Response Content Replace to expand.
 - a. In the **Content String** field, enter the content to look for in server responses. This is the text to be replaced.
 - b. In the **New String** field, enter the content that will be used to replace the original content.

Quotation marks are not required, even if either or both strings contains blank spaces.
 - c. Click **Add**.
5. Select or enter values for any additional template options you want to use. The remaining sections in this section describe the fields for configuring each option.
6. Click **OK**. The template appears in the ADC template list.

CLI Configuration

These commands configure replacement of “Welcome to A10” with “Greetings from A10!”:

```
ACOS(config)# slb template http http1
ACOS(config-http)# response-content-replace "Welcome to A10" "Greetings
from A10!"
```

Content Compression

Most types of real servers are able to compress media (content) before sending it to clients. Compression reduces the amount of bandwidth required to send content to clients. Although compression optimizes bandwidth, compression also can sometimes actually hinder overall website performance, if the real servers spend a lot of their CPU resources performing the compression.

To maximize the benefits of content compression, you can enable the ACOS device to perform compression for the real servers.

Compression is disabled by default. When it is enabled, the ACOS device compresses media of types “text” and “application” by default. The ACOS device can be configured to compress additional media types. The device can also be configured to exclude specific media types and even specific URIs from compression.

- Compression is supported only for HTTP and HTTPS virtual ports.
- Compression is not supported for fast-HTTP virtual ports.

Accept-Encoding Field

An HTTP request from clients usually contains an Accept-Encoding field in the header. This field indicates to the real server whether the client is willing to accept compressed content.

If compression is enabled on the real server, the real server will compress content before sending it to a client, if the client’s request contains the Accept-Encoding field with the “compress” value for the requested content type.

By default, when you enable compression on the ACOS device, the device removes the entire

Accept-Encoding field from the request before sending the request to the server. As a result, the server does not compress the content before sending it in the reply. ACOS compresses the content, then sends the reply with the compressed content to the client.

If you still want the server to compress some content, you can configure the ACOS device to leave the Accept-Request field unchanged. In this case, compression is performed by the real server instead of the ACOS device, if the server is configured to perform the compression. ACOS can still compress content that the real server does not compress.

Compression Level

ACOS supports compression level 1-9. Each level provides a higher compression ratio, beginning with level 1, which provides the lowest compression ratio. Higher compression ratios result in smaller file size after compression. Higher compression levels also require more CPU processing, so performance can be affected.

Default compression level is 1; this provides fastest compression speed but with lowest compression ratio.

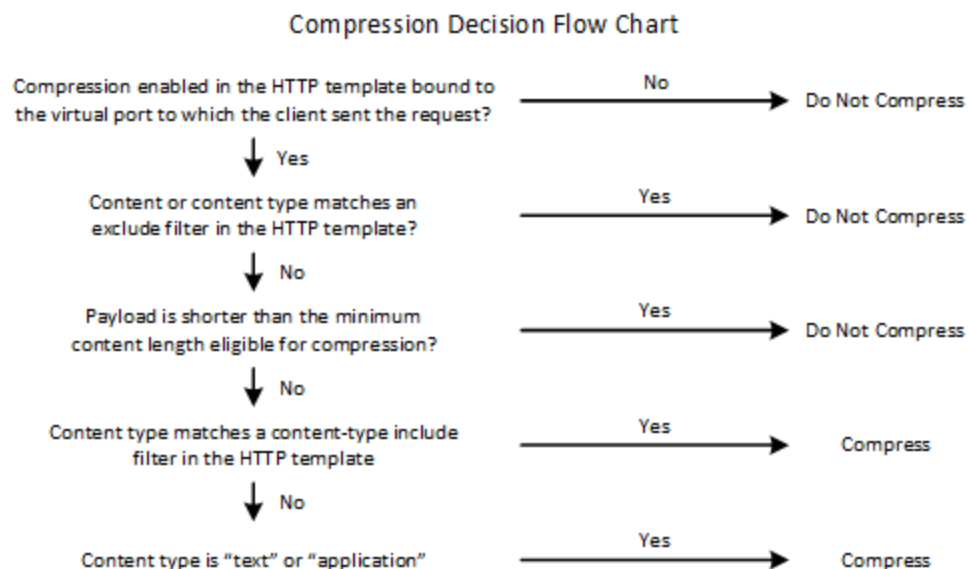
The actual performance impact of a given compression level depends on the content being

compressed. For example, if the content has a lot of repeated string patterns (for example, XML files), compression is faster than it is for content with few repeated string patterns (for example, graphics).

How ACOS Determines Whether to Compress a File

ACOS uses the following process to determine whether to compress a file before sending it to a client.

Figure 29 : Content Compression



If the ACOS device is configured to leave the Accept-Encoding field unchanged, and the real server has already compressed the file, the ACOS device forwards the compressed file without recompressing it.

CLI Configuration

This topic describes how to configure the content compression

These commands configure an HTTP template (http-compress) that uses compression level 5 for files with media type "application" or "image". Files of media type "application/zip" are excluded from compression.

```

ACOS(config)# slb template http http-compress
ACOS(config-http)# compression enable
ACOS(config-http)# compression level 5
ACOS(config-http)# compression content-type image
ACOS(config-http)# compression exclude-content-type application/zip
ACOS(config-http)# exit
  
```

Show Commands

These command displays HTTP compression statistics. The counters are in bytes and apply to all HTTP compression configured in all HTTP templates on the ACOS device.

- To view the gzip compression ratio and when gzip is prioritized, use the following show command:

```
ACOS(config)# show slb compression vip1 80 gzip
Virtual port name:                vip1:80:http
Virtual port number:              10.214.1.100:80
port 80
Number of requests compressed via gzip:    0
Number of requests NOT compressed via gzip: 0
Ratio of requests compressed via gzip:     0%
Data into gzip compression engine:        0.00B
Data out of gzip compression engine:       0.00B
Data Saved via gzip:                     0.00B
Gzip compression ratio:                  0.00%
```

- To view the total compression statistics , use the following show command:

```
ACOS(config)# show slb http-proxy
Total
-----
.....
Close on DDoS                0
Tot data before gzip compress 0
Tot data after gzip compress  0
Tot data before brotli compress 0
Tot data after brotli compress 0
.....
```

- To view the hardware-based compression statistics, use the following command:

```
ACOS(config)# slb common
ACOS(config-common)# hw-compression
ACOS(config-common)# exit
ACOS(config)# show slb hw-compression
Hardware compression device is installed.
Hardware compression module is enabled.
Total
```

```
-----  
total request count      177157  
total submit count      177157  
total response count    177157  
total failure count      0  
last failure code        0  
compression queue full  0  
max queued request count 84  
max queued submit count  68
```

GUI Configuration

This topic describes how to configure the content compression using GUI.

Navigate to the configuration page to create a new HTTP template:

1. Navigate to **ADC >> Templates >> L7 Protocols** tab.
2. Click **+ Create** and select **HTTP** from the drop-down menu.
3. Enter a name for the HTTP template.
4. Scroll to Compression and click to expand and display a set of additional fields.
5. To add more content types to be compressed:
 - a. In the Compression Content Type field, enter the string for a content type to compress.
 - b. Click **Add**.
 - c. Repeat step a and step b for each type of content to compress.
6. Click **OK**.
7. If your ACOS device supports hardware-based compression, enable the feature:
 - a. Navigate to **ADC >> SLB >> Global** tab.
 - b. Click the checkbox next to **Hardware Compression**.
 - c. Click **Update**.

Hardware Compression is not available when the ACOS device does not contain a compression module.

Brotli Compression

Brotli ([RFC 7932](#)) is a lossless compression technique that compresses data utilizing a combination of the LZ77 algorithm, Huffman coding, and 2nd order context modeling.

- LZ77 algorithm: LZ77 algorithm compresses data by replacing repeated occurrences of data with references to a single copy of data that existed earlier in the uncompressed data stream.
- Huffman coding: Huffman coding/algorithm uses a variable-length encoding strategy to assign binary codes to characters based on their frequency of occurrence in the given text. The character that appears the most frequently is allocated the smallest code, while the character that appears the least frequently is assigned the largest code.
- 2nd order context modeling: 2nd order context modeling is used to achieve higher density data by reusing the entropy codes, larger memory window of the past data, and the joint distribution codes.

Brotli compression is similar to gzip compression, however, they are smaller and faster than gzip. Unlike gzip, source files are preserved by default. Streams compressed by Brotli have the content-encoding response header as "br."

Web servers and content delivery networks use Brotli to compress HTTP data, resulting in faster page load and reduces bandwidth consumption. ACOS supports compression and decompression of brotli data stream.

The following functionalities are supported in this feature:

- Decodes/encodes the incoming HTTP traffic when the sliding window value of the compression is acceptable.
- Supports existing compression features under the HTTP template.
- Supports both HTTP/2 and HTTP 1.1 protocols (if the 'method-order' is specified under HTTP template).
- Extends ram cache support for HTTP response caching when it is compressed in Brotli format.
- Supports aFlex commands.

For detailed information about compression level and how ACOS determines to compress a file, refer [Content Compression](#).

CLI Configuration

This topic describes how to configure the Brotli compression under HTTP template using CLI.

These commands configure an HTTP template (http-compress) that uses brotli compression level 5.

```
ACOS(config)# slb template http http-compress
ACOS(config-http)# compression enable
ACOS(config-http)# compression brotli-level 5
ACOS(config-http)# compression brotli-sliding-window 2
ACOS(config-http)# compression method-order "gzip brotli"
ACOS(config-http)# exit
```

Show Commands

These commands display the HTTP compression statistics. The counters are in bytes and apply to all HTTP compression configured in all HTTP templates on the ACOS device.

- To view the brotli compression ratio and when brotli is prioritized, use the following show command:

```
ACOS(config)# show slb compression vip1 80 brotli
Virtual port name:                vip1:80:http
Virtual port number:              10.214.1.100:80
port 80
Number of requests compressed via brotli: 0
Number of requests NOT compressed via brotli: 0
Ratio of requests compressed via brotli: 0%
Data into brotli compression engine: 0.00B
Data out of brotli compression engine: 0.00B
Data Saved via brotli:            0.00B
Brotli compression ratio:         0.00%
```

- To view the total compression statistics, use the following show command:

```
ACOS(config)# show slb http-proxy
Total
-----
.....
```



```

Close on DDoS                0
Tot data before gzip compress 0
Tot data after gzip compress  0
Tot data before brotli compress 0
Tot data after brotli compress 0
.....

```

- To view the HTTP requests compression statistics of the virtual server, use the following show command:

```

ACOS(config)# show slb virtual-server vip1 port 80 http
...
Data into gzip compression engine                0
Data out of gzip compression engine               0
Data into brotli compression engine               0
Data out of brotli compression engine             0
Data into compression engine                     0
Data out of compression engine                   0
Number of requests compressed via gzip            0
Number of requests NOT compressed via gzip        0
Number of requests compressed via brotli          0
Number of requests NOT compressed via brotli      0
Number of requests compressed                    0
Number of requests NOT compressed                0
...

```

GUI Configuration

This topic describes how to configure the Brotli compression using GUI.

Navigate to the configuration page to create a new HTTP template:

1. Navigate to **ADC >> Templates >> L7 Protocols** tab.
2. Click **+ Create** and select **HTTP** from the drop-down menu.
3. Enter a name for the HTTP template.
4. To add brotli-related settings, expand **Compression** section and configure the following options:

- Brotli Compression Level
 - Brotli Compression Sliding Window
 - Method Order
5. To add more content types to be compressed:
 - a. In the Compression Content Type field, enter the string for a content type to compress.
 - b. Click **Add**.
 - c. Repeat step a and step b for each type of content to compress.
 6. Click **OK**.

HTTP Requests Options

This section described various options to define how your server should handle the HTTP requests.

- IP insertion – Inserts client IP address into GET requests before sending requests to a real server. The address is added as a value to the X-ClientIP field by default. You can add the client address to a different field; for example: X-Forwarded-For. ([Client IP Insertion / Replacement](#).)
- Header insertion / erasure – Inserts a *field:value* pair into requests or responses, or deletes a header. ([Header Insertion / Erasure](#).)
- Disallow HTTP Methods - Disallow the HTTP methods such as GET, POST and drop the request when the condition matches. ([Disallow HTTP Methods](#).)

Header Insertion / Erasure

You can configure the ACOS device to insert, replace, or erase headers in client requests or server responses. Like other HTTP options, header insertion and erasure are configured using HTTP template options. Insert and delete options can be used in the same HTTP template.

An HTTP template can contain options to insert, replace, or erase a maximum of 8 headers. To configure ACOS to insert the client's IP address, see [Client IP Insertion / Replacement](#).

NOTE: ACOS does not allow an HTTP template with any of these header options to be bound to a fast-http virtual port. Likewise, ACOS does not allow any of the header options to be added to an HTTP template that is already bound to a fast-http virtual port.

Configuring Header Insertion / Replacement

To configure header insertion or replacement, specify the header (the *field:value* pair), and the insert or replace option. The insert / replace option can be one of the following:

- Insert-always – always inserts the *field:value* pair. If the request already contains a header with the same *field* name, the new *field:value* pair is added after the existing *field:value* pair. Existing headers are not replaced.
- Insert-if-not-exist – inserts the header only if the packet does not already contain a header with the same *field* name.
- Default behavior (neither of the options above) – inserts the header. If the packet already contains one or more headers with the specified *field* name, this option replaces the last header.

Effects of the Insert / Replace Options

Here are some examples of the effects of the insert / replace options: insert-always, insert-if-not-exist, and the default (no options). For these examples, assume that a client's request packet already contains the following Cookie headers: "Cookie: a=1" and "Cookie: b=2".

```
GET / HTTP/1.1
Host: www.example.com
Cookie: a=1
Cookie: b=2
...
```

Effect When insert-always Is Used

If you configure an HTTP template to insert “Cookie: c=3”, and you use the insert-always option, the client’s header is changed as follows:

```
GET / HTTP/1.1
Host: www.example.com
Cookie: a=1
Cookie: b=2
Cookie: c=3
...
```

Effect When insert-if-not-exist Is Used

If you configure an HTTP template to insert “Cookie: c=3”, and you use the insert-if-not-exist option, the client’s header is changed only if it does not contain any “Cookie” headers. Therefore, the client request in this example is unchanged:

```
GET / HTTP/1.1
Host: www.example.com
Cookie: a=1
Cookie: b=2
...
```

Effect When Default Behavior (Neither Option Above) Is Used

If you configure an HTTP template to insert “Cookie: c=3”, but you do not use either the insert-always or insert-if-not-exist option, the header is always inserted into the request. If the packet already contains a “Cookie” header, the header is replaced. If the packet contains multiple “Cookie” headers, the last one is replaced. Here is the result:

```
GET / HTTP/1.1
Host: www.example.com
Cookie: a=1
Cookie: c=3
...
```

GUI Configuration

1. Navigate to **ADC >> Templates >> L7 Protocols** tab.
2. Click **+ Create** and select **HTTP** from the drop-down menu.
3. Enter a name for the HTTP template.
4. Scroll to Header Insert and click to expand and display a set of additional fields.
5. To insert a request header:
 - a. In the Request Header Insert section, enter the *name:value* pair in the **Name** field.
 - b. By default, if a packet already contains one or more headers with the specified *field* name, the command replaces the first header. Optionally, to change this behavior, select one of the following options from the drop-down list next to the Name field:
 - **Insert Always** – ACOS always inserts the *field:value* pair. If the request already contains a header with the same *field* name, the new *field:value* pair is added after the existing *field:value* pair. Existing headers are not replaced.
 - **Insert If Not Exist** – ACOS inserts the header only if the packet does not already contain a header with the same *field* name.
 - c. Click **Add**.
6. To insert a response header, follow the same steps as those for inserting a request header, but use the “Response Header Insert” section.
7. Click **OK**. The HTTP template appears in the SLB template list.

CLI Configuration

To insert a header, use the `request-header-insert` command. See the “Command Line Interface Reference for ADC” for information about the command.

The following command configures an HTTP template that inserts “Cookie: c=3” into every HTTP request. If the request already contains “Cookie” headers, the first header is replaced.

```
ACOS(config)# slb template http replace-cookie
ACOS(config-HTTP template)# request-header-insert "Cookie: c=3"
```

The following command configures an HTTP template that always inserts “Cookie: c=3” into HTTP requests, but does not replace other “Cookie” headers. The “Cookie: c=3” header is added after any “Cookie” headers that are already present in the request.

```
ACOS(config)# slb template http add-cookie
ACOS(config-HTTP template)# request-header-insert "Cookie: c=3" insert-
always
```

The following command configures an HTTP template that inserts “Cookie: c=3” into HTTP requests, but only if the requests do not already have a “Cookie” header.

```
ACOS(config)# slb template http add-cookie-unless-present
ACOS(config-HTTP template)# request-header-insert "Cookie: c=3" insert-if-
not-exist
```

Configuring Header Erasure

The following sections show how to erase headers from HTTP requests or responses.

GUI Configuration

1. Navigate to **ADC >> Templates >> L7 Protocols** tab.
2. Click **+ Create** and select **HTTP** from the drop-down menu.
3. Enter a name for the HTTP template.
4. Scroll to Header Erase and click to expand and display a set of additional fields.
5. To erase a request header:
 - a. In the Request Header Erase section, enter the field name (the *name* portion of the *name:value* pair) in the Name field.
 - b. Click **Add**.
6. To erase a response header, follow the same steps as those for erasing a request header, but use the “Response Header Erase” section.
7. Click **OK**. The HTTP template appears in the SLB template list.

CLI Configuration

Use the **response-header-erase** command to remove the Set-Cookie header from HTTP responses:

```
ACOS(config)# slb template http http-erase-header  
ACOS(config-http)# response-header-erase Set-Cookie
```

Client IP Insertion / Replacement

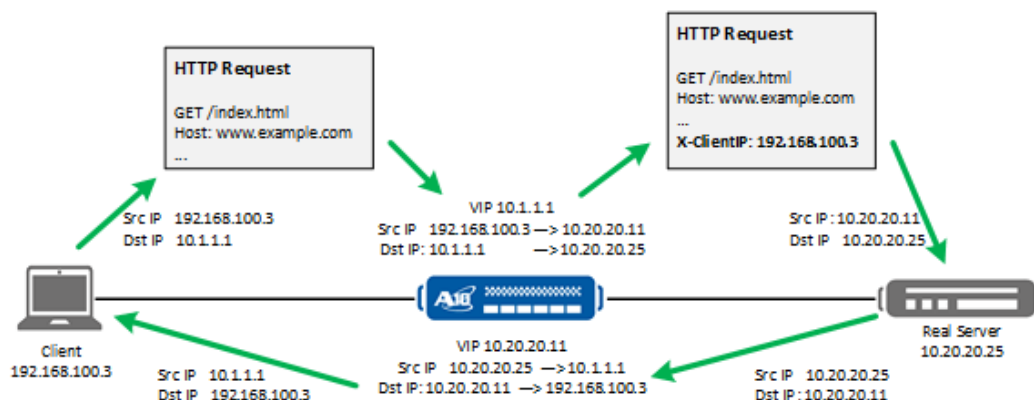
Many websites find it useful to know the IP addresses of clients. When the default Network Address Translation (NAT) settings for SLB are used, the source IP address of a request received by the ACOS device continues to be the source IP address when the ACOS device sends the request to a real server. The real server therefore knows the IP addresses of its clients. (An example is shown in [SLB NAT](#).)

In configurations where IP source NAT is enabled for SLB, the client IP address is not the source IP address in requests received by the real server. Instead, the source IP address of the request is the address into which the ACOS device translates the client's IP address.

To add the client's IP address back to the request, you do not need to change the network configuration or NAT settings. Instead, you can simply enable the ACOS device to insert the client's IP address into the header of the client's GET request before sending the request to a real server.

[Figure 30](#) shows an example of client IP insertion.

Figure 30 : Client IP Insertion



In this example, SLB source NAT changes the source address of the client's GET request from 192.168.100.3 to 10.20.20.11. However, the client's source IP address is preserved within the HTTP header of the request, by inserting the address into the X-ClientIP field.

This option inserts the client IP address into the X-ClientIP field by default. However, another field name can be specified. For example, configuring the option to insert the client IP address to the X-Forwarded-For field.

To insert HTTP header fields with other value types or to erase fields, see [Header Insertion / Erasure](#).

Replace Option

By default, the client IP address is appended to addresses already in the target header field. You can configure the ACOS device to replace any addresses that are already in the field.

Without this option, the client IP address is appended to the lists of client IP addresses already in the header. For example, if the header already contains "X-Forwarded-For:1.1.1.1", the field:value pair becomes "X-Forwarded-For:1.1.1.1, 2.2.2.2".

If you enable replacement of the client IP addresses, the field:value pair becomes "X-Forwarded-For:2.2.2.2".

Configuring Client IP Insertion / Replacement

The following sections show how to enable client IP insertion / replacement.

Inserting the client-IP in to the TCP options header is configurable using an SLB TCP template.

1. Create an SLB TCP template and enable “insert-client-ip” on the template.

For Layer 4 virtual ports and Fast-HTTP, configure an SLB TCP template. For Layer 7 virtual ports, configure an SLB TCP-Proxy template.

2. Apply the template to the virtual ports for which you want to insert the client-IP into the traffic headers.

GUI Configuration

1. Navigate to **ADC >> Templates >> L7 Protocols** tab.
2. Click **+ Create** and select **HTTP** from the drop-down menu.
3. Enter a name for the HTTP template.
4. Scroll to Client IP Header Insert and click the checkbox to enable the option and display the name of the header field to which the client IP address will be added.
5. To change the field name, edit the name. Otherwise, leave the field name set to the default (X-ClientIP).
6. Optionally, to replace any client addresses that are already in the header, click the Replace Existing header checkbox. Without this option, the client IP address is appended to the lists of client IP addresses already in the header.
7. Click **OK**. The HTTP template appears in the SLB template list.

CLI Configuration

Example 1

The following commands implement the client IP insertion configuration shown in [Client IP Insertion](#).

The following commands configure the HTTP template:

```
ACOS(config)# slb template http insertclientip
ACOS(config-http)# insert-client-ip
```

```
ACOS(config-http) # exit
```

The following commands bind the HTTP template to virtual port 80:

```
ACOS(config) # slb virtual-server vs1 1.1.1.1
ACOS(config-slb vserver) # port 80 http
ACOS(config-slb vserver-vport) # template http insertclientip
```

Example 2

The following example configures an SLB TCP template and binds it to a port 80 TCP on the “proxy-server.”

The commands below create an SLB TCP template named “proxy-header” and configure it to insert the client-IP in the headers of forwarded traffic.

```
ACOS(config) # slb template tcp proxy-header
ACOS(config-l4 tcp) # insert-client-ip
ACOS(config-l4 tcp) # exit
```

The following commands apply the “proxy-header” template to the virtual server “proxy-server,” on port 80 TCP. All traffic coming through port 80 on “proxy-server” will have the client-IP inserted into the header when they are forwarded on.

```
ACOS(config) # slb virtual-server proxy-server 1.1.1.100
ACOS(config-slb vserver) # port 80 tcp
ACOS(config-slb vserver-vport) # template tcp proxy-header
ACOS(config-slb vserver-vport) # exit
ACOS(config-slb vserver) # exit
```

Configuring Client-IP Insertion

Inserting the client-IP in to the TCP options header is configurable using an SLB TCP template.

1. Create an SLB TCP template and enable “insert-client-ip” on the template.

For Layer 4 virtual ports and Fast-HTTP, configure an SLB TCP template. For Layer 7 virtual ports, configure an SLB TCP-Proxy template.

2. Apply the template to the virtual ports for which you want to insert the client-IP into the traffic headers.

Client-IP Insertion Extensive Support

ACOS provides expanded support for inserting the client's IP address into the TCP header. All modes of TCP, including Layer 4 TCP virtual ports, Layer 7 virtual ports, and fast-http are supported. Additionally, the client-IP can be inserted in the TCP header for IPv4 to IPv6, IPv6 to IPv6, and IPv6 to IPv4 traffic.

When client-IP Insertion is configured, the client's IP address is inserted into the TCP options. The options field is a maximum of 40 bytes. Other options may take priority over inserting the client's IP address, causing the client-IP to be omitted due to lack of room in the TCP header. This is most likely to occur in IPv6 to IPv6 or IPv6 to IPv4 traffic because the option length for an IPv6 address is longer. There is a counter that tracks the number of times the client-IP insertion is skipped due to lack of room in the TCP header.

Server Replies Options

URL Redirect Rewrite

ACOS can be configured to alter redirect messages sent by real servers. You can use redirect-rewrite options to make the following types of changes:

- URL change – You can change the URL before sending the redirect to the client. For example, if the real server redirects the client to `http://www.example1.com`, you change the URL to `http://www.example2.com` or `https://www.example2.com`.
- Secure redirection – You can change an unsecure redirect (HTTP) to a secure one (HTTPS). For example, if the real server redirects the client to `http://www.example1.com`, you change the URL to `https://www.example1.com`.

You can use one or both options.

Redirect-Rewrite Rule Matching

If a URL matches on more than redirect-rewrite rule within the same HTTP template, the ACOS device selects the rule that has the most specific match to the URL. For

example, if a server sends redirect URL 66.1.1.222/000.html, and the HTTP template has the redirect-rewrite rules shown below, the ACOS device will use the last rule because it is the most specific match to the URL:

```
slb template http 1
  redirect-rewrite match /00 rewrite-to http://66.1.1.202/a
  redirect-rewrite match /000.html rewrite-to /001.gif
  redirect-rewrite match 66.1.1.222/000.html rewrite-to
66.1.1.202/003.bmp
```

Configuring URL Redirect Rewrite

GUI Configuration

1. Navigate to **ADC >> Templates >> L7 Protocols** tab.
2. Click **+ Create** and select **HTTP** from the drop-down menu.
3. Enter a name for the HTTP template.
4. Scroll to Redirect Rewrite and click to expand and display a set of additional fields.
5. To change “http://” to “https://”:
 - a. Click **Use HTTPS**.
 - b. To change the SSL port number, edit the number in the Port field (the default is 443).
6. To change the URL:
 - a. In the Match List section, enter the URL string to be changed in the **URL Match** field.
 - b. In the **Rewrite To** field, enter the new URL.
7. Click **Add**.
8. Click **OK**. The HTTP template appears in the SLB template list.

CLI Configuration

This example configures the ACOS device to change unsecure URLs to secure URLs in redirect messages.

The following commands configure an HTTP template to redirect URLs beginning with “http://” to “https://” (URLs in the redirect messages are otherwise unchanged), then bind the HTTP template to virtual port 80:

```
ACOS(config)# slb template http secureredirect
ACOS(config-HTTP template)# redirect-rewrite secure
ACOS(config-HTTP template)# exit
ACOS(config)# slb virtual-server vs1 1.1.1.1
ACOS(config-slb vserver)# port 80 http
ACOS(config-slb vserver-vport)# template http secureredirect
```

Inserting HTTP Client Port Numbers in the HTTP Header

When an ACOS device forwards an HTTP packet to the server, the client port number is added to the HTTP header by adding the `insert-client-port` command in an HTTP template and binding the template to the HTTP virtual port. The command includes a **replace** option to replace content of an existing header that matches the configured name with the client’s port number. If no header name is specified, X-ClientPort is used as the default header name.

CLI Configuration

The following example configures the HTTP template:

```
ACOS(config)# slb template http insertclientport
ACOS(config-http)# insert-client-port MY_HEADER_NAME
ACOS(config-http)# exit
```

The following example binds the HTTP template to virtual port 80:

```
ACOS(config)# slb virtual-server vs1 1.1.1.1
ACOS(config-slb vserver)# port 80 http
ACOS(config-slb vserver-vport)# template http insertclientport
```

HTTP Protocol Compliance

ACOS provides HTTP protocol/request compliance check capabilities to defend from HTTP/1 and HTTP/2 smuggling attacks. HTTP request smuggling exploits differences in

how the front-end servers (proxies) and back-end servers execute requests from multiple senders.

You can configure HTTP protocol compliance checks/conditions and enable 'drop' action when the conditions are met. For example, ACOS will send a 400 response to the client-side before dropping the request.

The following conditions can be configured:

- **get-and-payload** - Configure the drop action if the HTTP/1 or HTTP/2 request has GET method and Transfer-Encoding (TE) or Content-Length (CL) header simultaneously.
- **h2up-content-length-alias drop** Configure the drop action if the content-length/content_length header appears in the HTTP/2 request.
- **h2up-with-host-and-auth** - Configure the drop action if the HTTP/2 request has both host header and :authority pseudo header.
- **h2up-with-transfer-encoding** - Configure the drop action if the HTTP/2 request includes TE header.
- **header-filter-rule** - [Configure the filter rules](#) for the HTTP headers content and apply the specified action. The traffic is filtered or dropped as per the defined rules.
- **malformed-h2up-header-value** - Configure the drop action if the header value of the HTTP/2 request includes \r or \n.
- **malformed-h2up-scheme-value** - Configure the drop action if the scheme value of HTTP/2 request is not HTTP/HTTPS.
- **multiple-content-length** - Configure the drop action if the HTTP/1 or HTTP/2 request has multiple CL headers.
- **multiple-transfer-encoding** - Configure the drop action if the HTTP/1 or HTTP/2 request has multiple TE headers.
- **transfer-encoding-and-content-length** - Configure the drop action if the HTTP/1 or HTTP/2 request has both the CL and TE headers.

CLI Configuration

```
ACOS(config) # slb template http abc
ACOS(config-http) # http-protocol-check
```

```
ACOS(config-http-http-protocol-check) # h2up-content-length-alias drop  
ACOS(config-http-http-protocol-check) # malformed-h2up-header-value drop
```

Configuring HTTP Header Filter Rules

The [HTTP Protocol Compliance](#) supports HTTP header filter rules configuration using 'header-filter-rule' option. The HTTP headers are filtered using full-text or regular expressions (regex). You can specify the header-name to be detected and enable the 'drop' action when the header-name matches the filter rules.

You can configure a maximum of five filter rules. When more than one rule matches an HTTP request, the rule with the lower sequence number is executed first. As a result, the lower the sequence number, the higher the priority when this condition occurs.

The 'header-filter-rule' has the following options, and each must be specified explicitly. Especially, the match-type and the action options must be configured before the header-name and header-value.

- header-filter-rule - Specify the sequence number or the filter rule ID.
- match-type - Specify the regular expression filter string to match. The match-type can be either 'full-text' or 'PCRE' compatible regex. For information about PCRE-compatible regex, see official *Perl Documentation*.
- action - Specify the 'drop' action.
- header-name - Specify the regular expression of the header-name.
- header-value - Specify the regular expression of the header-value.

CLI Configuration

```
ACOS(config) # slb template http abc  
ACOS(config-http) # http-protocol-check  
ACOS(config-http-http-protocol-check) # header-filter-rule 1
```

You can specify the value between 0 to 4.

```
ACOS(config-http-http-protocol-check-head...) # match-type full-text  
ACOS(config-http-http-protocol-check-head...) # action drop  
ACOS(config-http-http-protocol-check-head...) # header-name content-encoding
```

```
ACOS(config-http-http-protocol-check-head...) # header-value chunked
```


HTTP/2 Load Balancing

HTTP/2 is version 2 of the HTTP network protocol defined by [RFC 7540](#). The primary HTTP/2 difference from the base HTTP protocol is in the framing and transporting data between client and server. HTTP/2 optimizes the HTTP protocol due to the efficiency of multiplexing connections over the same connection, header field compression, and preemptive server-side push frames.

HTTP/2 protocol uses two new identifiers:

- h2 string - identifies the protocol in which HTTP/2 uses TLS.
- h2c string - identifies the protocol in which HTTP/2 runs without TLS over cleartext TCP.

NOTE: Starting with 5.2.1-P6 release, the HTTP/1 to HTTP/2 upgrade mechanism will not be supported as specified in [RFC 9113](#).

ACOS automatically upgrades to HTTP/2 if a client initiates an HTTP/2 connection.

CLI Configuration

ACOS supports upgrade headers and HTTP/2 traffic using `support-http2` command. This command allows you to configure an HTTP/2 port.

NOTE: If this command is not configured, then ACOS will not respond to HTTP/2 traffic and will ignore or strip upgrade headers.

Use the following guidelines to configure client-side and server-side HTTP/2 version:

- If the real server supports HTTP/2 is known, configure the `support-http2` command at the real server port level. In this case, the source-NAT is not required for end-to-end HTTP/2.
- If the real server supports HTTP/2 is unknown, do not configure the `support-http2` command at the real server port since the source-NAT is required for the

client connection and the associated HTTP/1.1 requests the real server to negotiate the HTTP/2 connection.

- If the real server supports HTTP/2 and source-NAT is required for the design architecture, multiple server connections are opened for each client connection.

The following example implements HTTP/2 for a real server that is configured for HTTP/2 and a virtual server configured for HTTP and HTTPS.

For HTTPS, the support-http2 command is configured and optionally source-NAT is configured for design requirements.

```
ACOS(config)# ip nat pool snat1 10.20.20.100 10.20.20.101 netmask /24
ACOS(config)# slb server s1 10.20.20.20
ACOS(config-real server)# port 80 tcp
ACOS(config-real server-node port)# support-http2
ACOS(config-real server-node port)# exit
ACOS(config-real server)# port 443 tcp
ACOS(config-real server-node port)# support-http2
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
ACOS(config)# slb service-group SG-1 tcp
ACOS(config-slb svc group)# member s1 80
ACOS(config-slb svc group-member:80)# exit
ACOS(config-slb svc group)# exit
ACOS(config)# slb service-group SG-2 tcp
ACOS(config-slb svc group)# member s1 443
ACOS(config-slb svc group-member:443)# exit
ACOS(config-slb svc group)# exit
ACOS(config)# slb template client-ssl c1
ACOS(config-client ssl)# certificate new.pem key new.key
ACOS(config-client ssl)# exit
ACOS(config)# slb template server-ssl ssl-templatel
ACOS(config-server ssl)# exit
ACOS(config)# slb virtual-server v1 10.10.10.10
ACOS(config-slb vserver)# port 80 http
ACOS(config-slb vserver-vport)# source-nat pool snat1
ACOS(config-slb vserver-vport)# service-group SG-1r
ACOS(config-slb vserver-vport)# exit
ACOS(config-slb vserver)# port 443 https
ACOS(config-slb vserver-vport)# source-nat pool snat1
ACOS(config-slb vserver-vport)# service-group SG-2
```

```
ACOS(config-slb vserver-vport)# support-http2
ACOS(config-slb vserver-vport)# template server-ssl ssl-template1
ACOS(config-slb vserver-vport)# template client-ssl c1
ACOS(config-slb vserver-vport)# exit
ACOS(config-slb vserver)# exit
ACOS(config)#
```

Consolitated Configuration Example

Use the following configuration steps for the real server, service group, and virtual server to enable HTTP/2.

1. Configure the real server and service groups:

```
slb server apache1 10.40.40.3
port 80 tcp
support-http2
port 443 tcp
support-http2
!
slb service-group apache-80 tcp
member apache1 80
!
slb service-group apache-443 tcp
member apache1 443
!
```

2. Configure the virtual server and virtual server ports for HTTP and HTTPS. The **source-nat auto** command uses the interface IP address for source-NAT. SSL templates are configured and the **support-http2** command is required for HTTP and HTTPS:

```
slb virtual-server test1 10.14.14.100
port 80 http
source-nat auto
service-group apache-80
port 443 https
support-http2
source-nat auto
service-group apache-443 tcp
```

```
template server-ssl ssl-templatel
template client-ssl c1
```

3. Configure the Apache Server to enable server push.:

```
#==set http2 server push
<FilesMatch "\.html$">
Header set Link "</index_files/css.css>; rel=preload; as=style "
```

4. The **show slb** command displays the Server Push statistics with two new fields Push Promise Frame Sent and Unexpected PUSH_PROMISE Frame. For example, the output showing HTTP/2 statistics (not all fields shown).

```
ACOS# show slb s1 | section s1
Curr HTTP2 Session 0
Peak HTTP2 Session 0
Total HTTP2 Session 0
ACOS# show slb s1 detail
DP0 DP1 Total
-----
Curr HTTP2 Sessions 0 0 0
Peak HTTP2 Sessions 0 0 0
Total HTTP2 Sessions 0 0 0
...
HTTP2 Proxy alloc Error 0 0 0
Push Promise Frame Sent 11
Unexpected PUSH_PROMISE Frame 1
...
```

HTTP/1.1 to HTTP/2 Communication Support

ACOS supports communication between HTTP/1.1 to HTTP/2 when HTTP/2 is configured on the server side, and the incoming traffic on the client side is HTTP/1.1. This capability is useful when you wish to have separate policies on the client side and the server side. This feature is enabled using the **support-http2 only** option under the real port and **server-support-http2-only** option under the HTTP template.

Use the following guidelines to configure the communication flow between client-side HTTP/1.1 and server-side HTTP/2:

1. Configure the SLB real server and real port.
2. Enable the `support-http2 only` option under real port.

NOTE: You can specify the `support-http2` option under the SLB real server port for server-side HTTP/2 to indicate that the server supports both HTTP/1.1 and HTTP/2. ACOS will send the HTTP/2 request with prior-knowledge to the server if the client request is HTTP/2.

3. Configure an SLB service group.
4. Add the real port to the service group.
5. Create the SLB HTTP template.
6. Configure the `server-support-http2-only auto-detect` Or `server-support-http2-only force` option.
7. Configure the SLB virtual server and virtual port.
8. Enable the `support-http2` option under virtual port.
9. Bind the service group and HTTP template to the virtual port.

Here,

- `support-http2 only` option indicates that the server can only handle HTTP/2 traffic.
- `server-support-http2-only auto-detect` option indicates that the virtual port will automatically utilize HTTP/2 for communicating with the server only when the `support-http2 only` option is configured in the real port and added to the service group that is bound to the virtual port.
- `server-support-http2-only force` option indicates that the virtual port will be forced to utilize HTTP/2 for communicating with the server in any scenario as long as the `support-http2 only` option is configured in the real port.

NOTE: These configurations are optional and has no effect on the existing configurations.

Known Limitation

This feature does not support SSLi virtual port.

CLI Configuration

This section explains the HTTP/1.1 to HTTP/2 configuration examples in the following scenarios:

Example 1

The following configuration is basic environment example.

In this example, the HTTP/1.1 to HTTP/2 communication flow work as follows:

- The **support-http2 only** is configured under the real port and the real port is added as the member of service-group.
- The **server-support-http2-only auto-detect** is configured in the HTTP template.
- The service-group is bound to the virtual port.
- The server-side will select this real port only for communication.

```
ACOS(config)# ip nat pool snat1 10.20.20.100 10.20.20.101 netmask /24
ACOS(config)# slb server s1 10.20.20.20
ACOS(config-real server)# port 80 tcp
ACOS(config-real server-node port)# support-http2 only
ACOS(config-real server-node port)# exit
ACOS(config-real server)# port 443 tcp
ACOS(config-real server-node port)# support-http2 only
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
ACOS(config)# slb service-group SG-1 tcp
ACOS(config-slb svc group)# member s1 80
ACOS(config-slb svc group-member:80)# exit
ACOS(config-slb svc group)# exit
ACOS(config)# slb service-group SG-2 tcp
ACOS(config-slb svc group)# member s1 443
ACOS(config-slb svc group-member:443)# exit
ACOS(config-slb svc group)# exit
ACOS(config)# slb template client-ssl c1
ACOS(config-client ssl)# certificate new.pem key new.key
ACOS(config-client ssl)# exit
ACOS(config)# slb template server-ssl ssl-template1
ACOS(config-server ssl)# exit
```

```
ACOS(config)# slb template http http-2
ACOS(config-client ssl)# server-support-http2-only auto-detect
ACOS(config-client ssl)# exit
ACOS(config)# slb virtual-server v1 10.10.10.10
ACOS(config-slb vserver)# port 80 http
ACOS(config-slb vserver-vport)# source-nat auto
ACOS(config-slb vserver-vport)# service-group SG-1
ACOS(config-slb vserver-vport)# exit
ACOS(config-slb vserver)# port 443 https
ACOS(config-slb vserver-vport)# source-nat auto
ACOS(config-slb vserver-vport)# service-group SG-2
ACOS(config-slb vserver-vport)# support-http2
ACOS(config-slb vserver-vport)# template http http-2
ACOS(config-slb vserver-vport)# template server-ssl ssl-template1
ACOS(config-slb vserver-vport)# template client-ssl c1
ACOS(config-slb vserver-vport)# exit
ACOS(config-slb vserver)# exit
ACOS(config)#
```

Example 2

The following configuration is environment-specific example that includes various configurations such as aflex, HTTP connection reuse, TCP-proxy template, and so on.

In this example, the HTTP/1.1 to HTTP/2 communication will flow as below:

- The **support-http2 only** is configured under the real port.
- The **server-support-http2-only force** is configured in the HTTP template.
- The server-side will select the real port through aFlex, HTTP template, and so on for communication.

```
ACOS(config)# ip nat pool snat1 10.20.20.100 10.20.20.101 netmask /24
ACOS(config)# slb server S1 2.2.2.120
ACOS(config-real server)# health-check-disable
ACOS(config-real server)# port 80 tcp
ACOS(config-real server-node port)# health-check-disable
ACOS(config-real server-node port)# support-http2 only
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
ACOS(config)# slb service-group sg_http tcp
```

```
ACOS(config-slb svc group)# member S1 80
ACOS(config-slb svc group-member:80)# exit
ACOS(config-slb svc group)# exit
ACOS(config)# slb service-group sg11 tcp
ACOS(config-slb svc group)# member S1 443
ACOS(config-slb svc group-member:443)# exit
ACOS(config-slb svc group)# exit
ACOS(config)# slb template connection-reuse multiplex-default
ACOS(config-conn reuse)# limit-per-server 0
ACOS(config-conn reuse)# timeout 60
ACOS(config-conn reuse)# keep-alive-conn preopen
ACOS(config-conn reuse)# exit
ACOS(config)# slb template tcp-proxy tcp-proxy-default
ACOS(config-tcp proxy)# idle-timeout 60
ACOS(config-tcp proxy)# half-open-idle-timeout 10
ACOS(config-tcp proxy)# exit
ACOS(config)# slb template http http-frp-ade
ACOS(config-http)# insert-client-ip X-Forwarded-For
ACOS(config-http)# request-header-erase X-ClientIP
ACOS(config-http)# keep-client-alive
ACOS(config-http)# server-support-http2-only force
ACOS(config-http)# exit
ACOS(config)# slb template http enhancement
ACOS(config-http)# server-support-http2-only force
ACOS(config-http)# exit
ACOS(config)# slb template client-ssl aflex-cl
ACOS(config-client ssl)# certificate new.pem key new.key
ACOS(config-client ssl)# exit
ACOS(config)# slb template server-ssl tls1.3-s1
ACOS(config-server ssl)# exit
ACOS(config)# slb virtual-server N1 1.1.1.121
ACOS(config-slb vserver)# port 80 http
ACOS(config-slb vserver-vport)# source-nat pool snat
ACOS(config-slb vserver-vport)# service-group sg_http
ACOS(config)# slb virtual-server vs 1.1.1.115
ACOS(config-slb vserver)# redistribution-flagged
ACOS(config-slb vserver)# port 80 http
ACOS(config-slb vserver-vport)# support-http2
ACOS(config-slb vserver-vport)# aflex aflex-http2-85-0k
ACOS(config-slb vserver-vport)# source-nat auto
```



```
ACOS(config-slb vserver-vport)# service-group sgt
ACOS(config-slb vserver-vport)# template connection-reuse multiplex-
default
ACOS(config-slb vserver-vport)# template http http-frp-ade
ACOS(config-slb vserver-vport)# template tcp-proxy tcp-proxy-default
ACOS(config-slb vserver-vport)# exit
ACOS(config-slb vserver)# port 443 https
ACOS(config-slb vserver-vport)# support-http2
ACOS(config-slb vserver-vport)# source-nat auto
ACOS(config-slb vserver-vport)# service-group sgtt
ACOS(config-slb vserver-vport)# template connection-reuse multiplex-
default
ACOS(config-slb vserver-vport)# template http enhancement
ACOS(config-slb vserver-vport)# template server-ssl tls1.3-s1
ACOS(config-slb vserver-vport)# template client-ssl aflex-cl
ACOS(config-slb vserver-vport)# template tcp-proxy tcp-proxy-default
ACOS(config-slb vserver-vport)# exit
ACOS(config-slb vserver)# exit
```

HTTP/2 SSL Offload without SNAT

When the client is HTTP2 and the server is HTTP1, the source NAT participation was required under the virtual port if the client-side initialize multiple streams on the same connection. However, ACOS also supports HTTP/2 SSL offloading capability, which does not require the source NAT participation.

The 'http2-client-no-snat' feature can utilize to dynamically adjust the max-concurrent-stream value when the source NAT is not configured under the virtual port, and the client-side is HTTP/2, and the server-side is HTTP1.

In this case, the max-concurrent-streams will be set to '1' until the server supports HTTP/2 and accepts other values.

CLI Configuration

```
ACOS(config)# slb template http test
ACOS(config-http)# http2-client-no-snat
```

Consolidated Configuration Example

```
ACOS(config)# slb server s1 121.116.10.26
ACOS(config-real server)# port 80 tcp
ACOS(config-real server-node port)# support-http2
ACOS(config-real server-node port)# exit
ACOS(config)# slb service-group SG tcp
ACOS(config-slb svc group)# member s1 80
ACOS(config-slb svc group-member:80)# exit
ACOS(config)# slb template http test
ACOS(config-http)# compression enable
ACOS(config-http)# http2-client-no-snat
ACOS(config)# slb virtual-server vs1 121.116.10.100
ACOS(config-slb vserver)# port 80 http
ACOS(config-slb vserver-vport)# service-group SG
ACOS(config-slb vserver-vport)# support-http2
ACOS(config-slb vserver-vport)# template http test
ACOS(config-slb vserver-vport)# exit
ACOS(config-slb vserver)# exit
ACOS(config)#
```

HTTP Load Balancing to Proxy Servers

This section describes HTTP Load Balancing to Proxy Servers.

The following topics are covered:

Overview	248
Configuring HTTP Load Balancing to Proxy Servers	248
Configuring HTTP Headers to Forward to Proxy Servers	249

Overview

The ACOS external service module has fixed headers for URL filtering when it forwards requests to proxy servers. Besides these fixed headers, you also can specify a set of HTTP request-headers when forwarding HTTP packets from the client to the proxy servers. If there are multiple headers with the same name from the client, then only the first header instance will be forwarded.

The URL Filter server's HTTP module parses the client request and saves the results in the corresponding data structure. ACOS inserts the configured header when it forwards the HTTP request to the proxy server. If the response from the proxy server is good, then ACOS connects to the destination server. If the response from the proxy server is bad, then ACOS closes the connection.

To specify the HTTP request-headers to be sent to the proxy server, use an SLB “external-service” template.

- Only GET and POST methods are forwarded by the SLB “external-service” template, so only these methods will forward the configured request-headers to the proxy servers.
- A maximum of 16 HTTP headers can be forwarded. One HTTP header only can be 1036 bytes, including the HTTP header name and HTTP header element. Anything longer than that will be truncated at 1036 bytes.
- The maximum allowed length for HTTP header name based on HTTP1 traffic is 8191 bytes. The previous limit of 64 bytes still applies to configuration under the HTTP template.
- If there are multiple headers with the same name from the client, then only the first header instance will be forwarded.

Configuring HTTP Load Balancing to Proxy Servers

The number of HTTP headers that ACOS can process can be increased up to 255 by entering the `max-http-header-count` command from SLB common configuration mode.

The default value is 90, and a value between 90 and 255 can be entered.

Increasing the HTTP Header Processing Capacity (CLI Example)

This example increases the number of HTTP headers that ACOS can process to 255.

```
ACOS(config)# slb common
ACOS(config-common)# max-http-header-count 255
```

Configuring HTTP Headers to Forward to Proxy Servers

To configure HTTP headers to forward to proxy servers, you need to:

1. Create an SLB External-service template.
2. Specify, in the template, the request-headers that you want forwarded.

CLI Configuration

The following example enables header request forwarding within an SLB External-service template named “header-forwarding” for the headers “user-agent,” “date,” and “cookie.”

```
ACOS(config)# slb template external-service header-forwarding
ACOS(config-external-service)# request-header-forward cookie
ACOS(config-external-service)# request-header-forward DATE
ACOS(config-external-service)# request-header-forward User-Agent
```

Responding to Server Selection Failure

The following topics are covered:

Overview	250
TCP Reset	250
Gateway Timeout Error	253

Overview

ACOS provides options to clients for handling server selection failures. A server selection failure can occur as a result of any of the following conditions:

- Server or port connection limit is reached
- Server or port connection rate limit is reached
- Client in a PBSLB black/white list reaches its connection limit
- The `def-selection-if-pref-failed` option is disabled and SLB is unable to select a server for any reason
- All servers are down

The `def-selection-if-pref-failed` command enables ACOS to check for an available server in other service groups if all the servers are down in the first service group selected by SLB. The following options configure responses to client requests in case a server selection fails.

TCP Reset

ACOS provides an option to send a TCP reset (RST) to clients if server selection fails. The `reset-on-server-selection-fail` option can be enabled at either of the following levels:

- Service group
- Virtual port

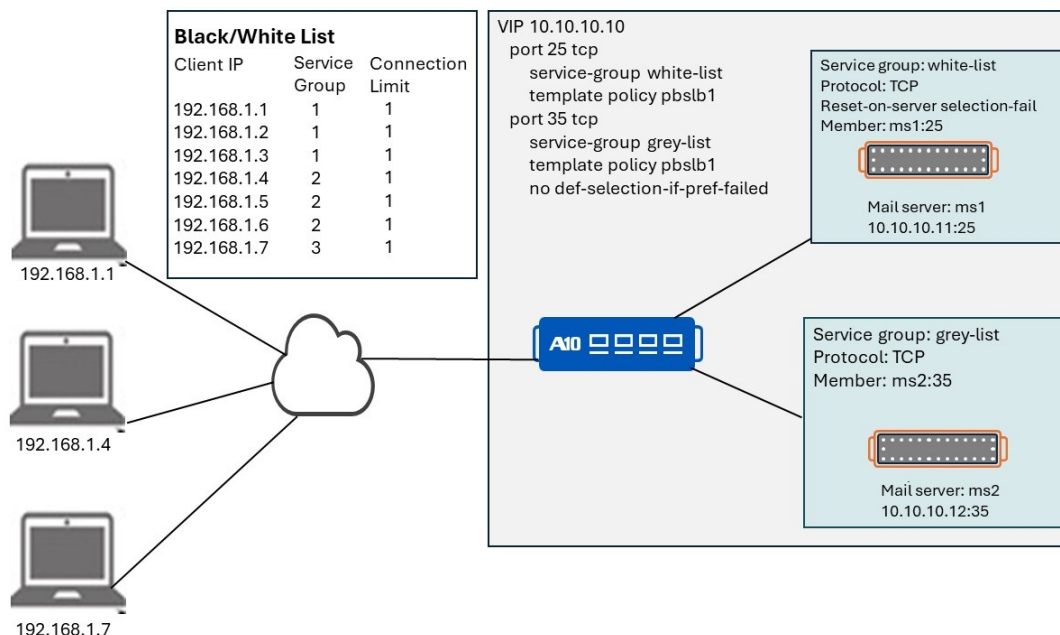
Enabling the `reset-on-server-selection-fail` option at the service-group level allows selective use of the option based on service group. [Figure 31](#) shows an example of a Policy-Based SLB (PBSLB) solution that uses the `reset-on-server-selection-fail` option.

The TCP template `reset-rev` option can also be used to send a RST to clients. In ACOS releases prior to 2.2.2, the `reset-rev` option would send a RST in response to a server selection failure. In ACOS Release 2.2.2 and later, the new `reset-on-server-selection-fail` option must be used instead.

When "reset-rev" is configured in a `tcp` or `tcp-proxy` template that is bound to a virtual port when the server is disabled or down, RST is not always immediately sent to the client:

- If there is an active session (ACOS is processing traffic and detects that the server for the session the traffic matches, is disabled or goes down), the RST is sent to client immediately.
- If the server is detected as disabled or down while the session is idle, RST is sent to the client after a 30 to 90 second delay. However, if "del-session-on-server-down" was configured in the SLB port template, RST is sent immediately.

Figure 31 : PBSLB Used With reset-on-server-selection-fail Option



This deployment categorizes clients as follows:

- White-list clients
- Grey-list clients

In this solution, if a white-list client exceeds the connection limit specified in the black/white list, the ACOS device sends a RST to the client. However, if a grey-list client exceeds its connection limit, the ACOS device drops the connection, instead of sending a RST.

To implement this solution, a separate service group is configured for each client category. In the black/white list, each client is assigned to one of the service groups, according to the client's category. For example, client 192.168.1.1 is a white-list client, and is therefore assigned to the white-list service group.

To configure the ACOS device to send a RST to white-list clients upon server selection failure, but not to grey-list clients, the reset-on-server-selection-fail option is used in the white-list service group only. The default PBSLB action; drop, is used for the other service group.

The virtual ports to which clients will send mail traffic are bound to the respective service groups. In addition, the def-selection-if-pref-failed option is disabled for the

grey-list service-group. This option must be disabled so that the ACOS device does not attempt to use other configuration areas of the system to select a server, if SLB is unable to select a server.

A policy template identifies the black/white list and service group assignments. It is bound to the virtual ports.

While the example uses a separate server for each client category, traffic for all clients can be sent to a single server. The solution uses a separate service group for each client category, enabling the `reset-on-server-selection-fail` option in the white-list service group.

Gateway Timeout Error

ACOS provides an error handling mechanism when the destination servers are down. This prevents client traffic blackholing and ensures that clients receive a response even in cases of server unavailability. ACOS selects a service group based on the URL or domain name in the client's GET request. When the `when-server-selection-failed send-504` option is enabled, ACOS returns 504 Gateway Timeout error when server selection fails. It indicates that the server is not reachable and cannot complete the request.

NOTE:	This command is exclusive to the <code>reset-on-server-selection-fail</code> command.

CLI Configuration

The following configuration enables ACOS to send the 504 Gateway Timeout error to client's GET request when server selection fails.

1. Configure a real server and port. Ensure that health check is enabled on the port so that it informs about server failure. It is enabled by default:

```
ACOS(config)#slb server rs1 10.20.20.1
ACOS(config-real server)#port 80 tcp
```

2. Configure service group.

```
ACOS(config)#slb service-group sg-tcp
```

```
ACOS(config-slb svc group)# member rsl 80
```

3. Configure virtual-server, virtual port, and the response to server selection failure.

```
ACOS(config)#slb virtual-server vs 10.10.20.100  
ACOS(config-slb vserver)#port 80 http  
ACOS(config-slb vserver-vport)#service-group sg-tcp  
ACOS(config-slb vserver-vport)#when-server-selection-failed send-504
```

SIP Load Balancing

This section describes Session Initiation Protocol (SIP) load balancing and how to configure it. You can configure load balancing for SIP over UDP or SIP over TCP/TLS.

The following topics are covered:

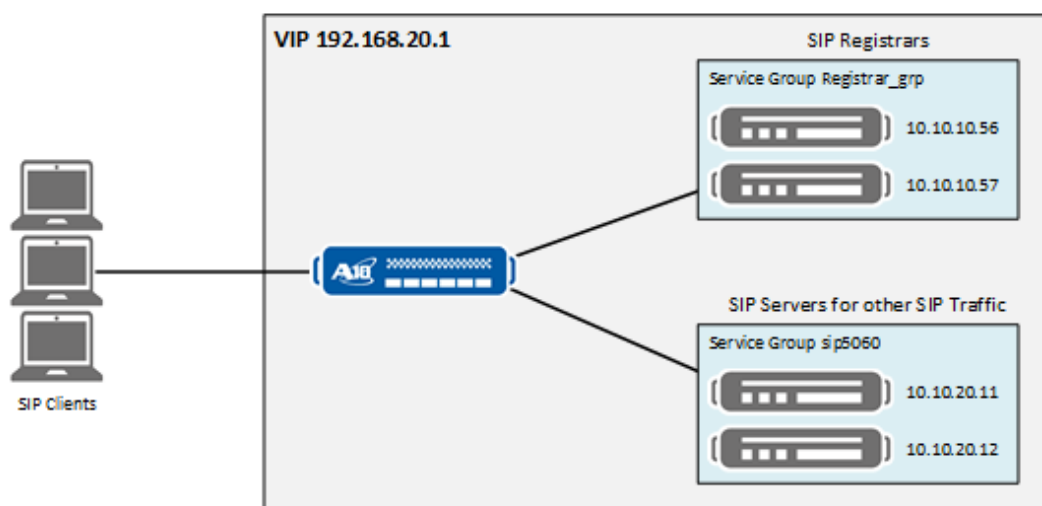
Load Balancing for SIP over UDP	256
Load Balancing for SIP over TCP/TLS	262

Load Balancing for SIP over UDP

ACOS devices support SIP load balancing. SIP load balancing balances SIP registration messages from clients across a service group of SIP Registrar servers. SIP load balancing enables you to offload registration processing from other SIP servers so those servers can more efficiently process other SIP traffic.

[Figure 32](#) shows an example of a SIP load balancing configuration. The commands to implement this configuration are shown in [Configuring Load Balancing for SIP over UDP](#).

Figure 32 : SIP Load Balancing



Application Layer Gateway Support

You can enable ALG support for SIP load balancing.

The status of ALG support does not affect address translation at the IP layer. Address translation at the IP layer is still performed, if applicable, even if ALG support is disabled.

Configuring Load Balancing for SIP over UDP

The following sections provide instructions for configuring Load Balancing for SIP over UDP.

- [Configuring Load Balancing for SIP over UDP \(Procedure\)](#)
- [GUI Configuration](#)
- [CLI Configuration](#)

Configuring Load Balancing for SIP over UDP (Procedure)

To configure load balancing for SIP over UDP:

1. Configure SIP health monitors using the SIP health method.
2. Configure a real server for each SIP Registrar server, add the SIP port to the server, and assign the SIP health monitor to the port.
3. Configure a real server as a proxy for each SIP server that will handle SIP messages other than registration messages. Add the SIP port to each server. The SIP port can be the same on the Registrar servers and these proxy servers. The ACOS selects a service group based on the message type.
4. Configure a service group for the Registrar servers and add them to the group.
5. Configure a service group for the other SIP servers and add them to the group. This is the SIP proxy group.
6. Configure a SIP template to redirect all SIP registration messages to the SIP Registrar service group.
7. Configure a virtual server containing the SIP port and bind the port to the SIP proxy group. Add the SIP proxy service group and the SIP template to the port.

GUI Configuration

The procedure includes the following steps:

To configure a SIP health monitor for the Registrar servers

1. Navigate to **ADC >> Health Monitors**.
2. Click **Create**.
3. In the Name field, enter a name for the health monitor.
4. Click the **Method** type drop-down menu and select SIP.
5. To send health checks to the default SIP port (5060), leave the port unchanged.

Otherwise, to send the request to a different port, edit the port number in the SIP Port field.
6. Select the Register checkbox to send a REGISTER request. (By default, an OPTION request is sent instead.)
7. For SIP over TCP, select the Use TCP checkbox.
8. To check the reply for specific response codes, enter them in the Expect Response Code field.
9. Click **Create Monitor**. The new SIP health monitor appears in the Health Monitor table.

To configure a real server for the SIP Registrar server

1. Navigate to **ADC >> SLB >> Servers** tab.
2. Click **Create**. The Create Server page appears.
3. In the Name field, enter a name for the Registrar server.
4. Click the **Type** radio button, and then enter the IP address of the server in the field below.
5. In the Port section, click **Create**.
6. In the Port field, enter "5060" for the SIP port number.
7. Click the **Protocol** drop-down list, and then select UDP.
8. For the Health Check radio button, select Monitor, and in the health monitor drop-down menu that appears below, select the SIP health monitor you just created.
9. Click **Create**. The port appears in the Port list.

Use the same steps to configure a real server as a proxy for each SIP server that handles SIP messages other than registration messages.

To configure a service group for the Registrar servers and add them to the group

1. Navigate to **ADC >> SLB >> Service Groups** tab.
2. Click **Create**. The Create Service Group page appears.
3. In the Name field, enter a name for the service group.
4. In the Protocol drop-down list, select UDP.
5. In the Port section, click Create then click the Server drop-down list and select the SIP Registrar server.
6. In the Port field, enter the SIP port number “5060”.
7. Click **Create**.
8. Repeat this process to add each Registrar server to the group.
9. Click **Update**. The service group appears in the service group table.
10. Use the same steps to configure a service group for the other SIP servers and add them to the group. This is the SIP proxy group.

To configure a SIP template to redirect SIP registration messages to SIP Registrar service group

1. Navigate to **ADC >> Templates >> L7 Protocols** tab.
2. Click **Create**.
3. Select SIP from the drop-down menu that appears.
4. Enter a name for the template in the Name field.
5. Optionally, erase, insert, or replace text in the SIP header.
6. Click the **Registrar Service Group** drop-down menu, and select the registrar service group.
7. Click **OK**. The new SIP template appears in the SIP template table.

To configure a virtual server for the SIP proxy:

1. Navigate to **ADC >> SLB >> Virtual Servers** tab.
2. Click **Create**.
3. In the Name field, enter a name for the virtual server.
4. Select the Address Type radio button (IPv4 or IPv6), and then enter the IP address to which clients will send SIP Registration messages.
5. In the Virtual Port section, click **Create**.
6. From the page that appears, click the Protocol drop-down menu and select SIP.
7. In the Port field, enter the SIP port number, "5060".
8. Click the **Service Group** drop-down menu and select the SIP service group you created above for non-registration traffic.
9. Click **Templates** to expand the template configuration options.
10. Click the **Template SIP** drop-down menu, and select the SIP template you just created.
11. Click **Create**. The port appears in the Port list for the virtual server.
12. Click **Update**. The virtual port appears in the virtual server table.

CLI Configuration

This example implements the SIP load balancing configuration shown in [SIP Load Balancing](#).

1. The following commands configure the SIP health monitors:

```
ACOS(config)# health monitor sip_monitor
ACOS(config-health:monitor)# method sip
ACOS(config-health:monitor)# exit
ACOS(config)# health monitor sipreg_monitor
ACOS(config-health:monitor)# method sip register
ACOS(config-health:monitor)# exit
```

2. The following commands configure the Registrar servers:

```
ACOS(config)# slb server Registrar1 10.10.10.56
ACOS(config-real server)# port 5060 udp
ACOS(config-real server-node port)# health-check sipreg_monitor
```



```
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
ACOS(config)# slb server Registrar2 10.10.10.57
ACOS(config-real server)# port 5060 udp
ACOS(config-real server-node port)# health-check sipreg_monitor
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
```

3. The following commands configure the SIP proxy servers:

```
ACOS(config)# slb server Proxy3 10.10.20.11
ACOS(config-real server)# port 5060 udp
ACOS(config-real server-node port)# health-check sip_monitor
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
ACOS(config)# slb server Proxy4 10.10.20.12
ACOS(config-real server)# port 5060 udp
ACOS(config-real server-node port)# health-check sip_monitor
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
```

4. The following commands configure the service groups:

```
ACOS(config)# slb service-group Registrar_gp udp
ACOS(config-slb svc group)# member Registrar1 5060
ACOS(config-slb svc group-member:5060)# exit
ACOS(config-slb svc group)# member Registrar2 5060
ACOS(config-slb svc group-member:5060)# exit
ACOS(config-slb svc group)# exit
ACOS(config)# slb service-group sip5060 udp
ACOS(config-slb svc group)# member Proxy3 5060
ACOS(config-slb svc group-member:5060)# exit
ACOS(config-slb svc group)# member Proxy4 5060
ACOS(config-slb svc group-member:5060)# exit
ACOS(config-slb svc group)# exit
```

5. The following commands configure the SIP template:

```
ACOS(config)# slb template sip Registrar_template
ACOS(config-sip)# registrar service-group Registrar_gp
ACOS(config-sip)# client-request-header insert max-Forwards:22
ACOS(config-sip)# client-request-header erase Contact
```

```
ACOS(config-sip) # exit
```

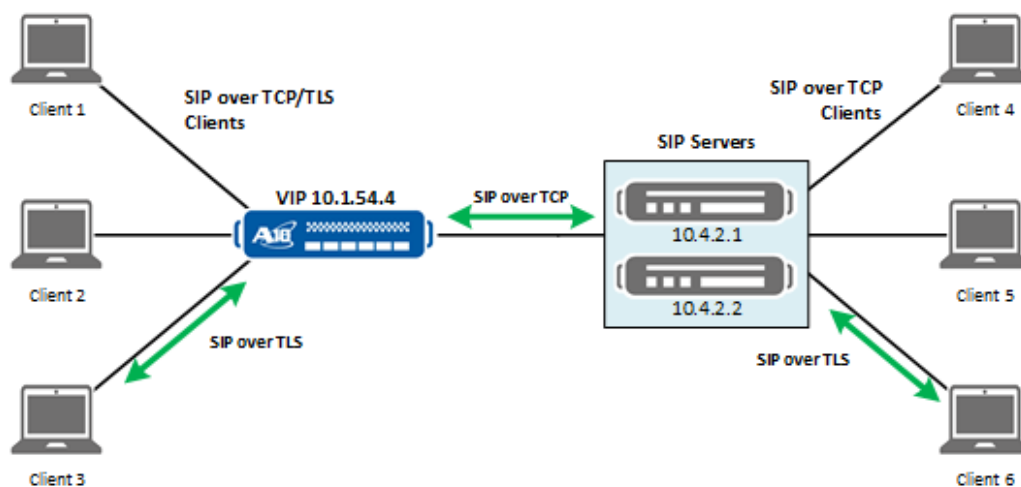
6. The following commands configure the VIP for the SIP registrar:

```
ACOS(config) # slb virtual-server sip1 192.168.20.1
ACOS(config-slb vserver) # port 5060 sip
ACOS(config-slb vserver-vport) # service-group sip5060
ACOS(config-slb vserver-vport) # template sip Registrar_template
ACOS(config-slb vserver-vport) # exit
ACOS(config-slb vserver) # exit
```

Load Balancing for SIP over TCP/TLS

SIP over TCP/TLS enables an ACOS device as a secure SIP proxy for SIP servers. [Figure 33](#) displays an example.

Figure 33 : SIP over TCP / TLS

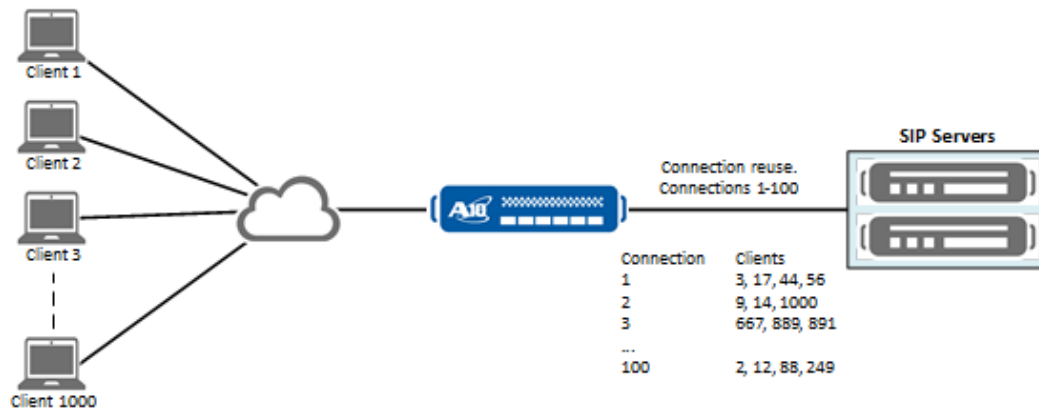


SIP clients send secure SIP requests over TLS. Requests are addressed to a VIP configured on the ACOS device. The ACOS device forwards the requests to the SIP servers over TCP. Likewise, when the ACOS device receives SIP traffic from the SIP servers, the ACOS device forwards the traffic to the appropriate clients over TLS.

SIP Multiplexing

You can use the ACOS device to multiplex SIP connections. This is useful in cases where the SIP servers do not have enough capacity to maintain separate connections for each SIP client. [Figure 34](#) shows an example.

Figure 34 : SIP Multiplexing

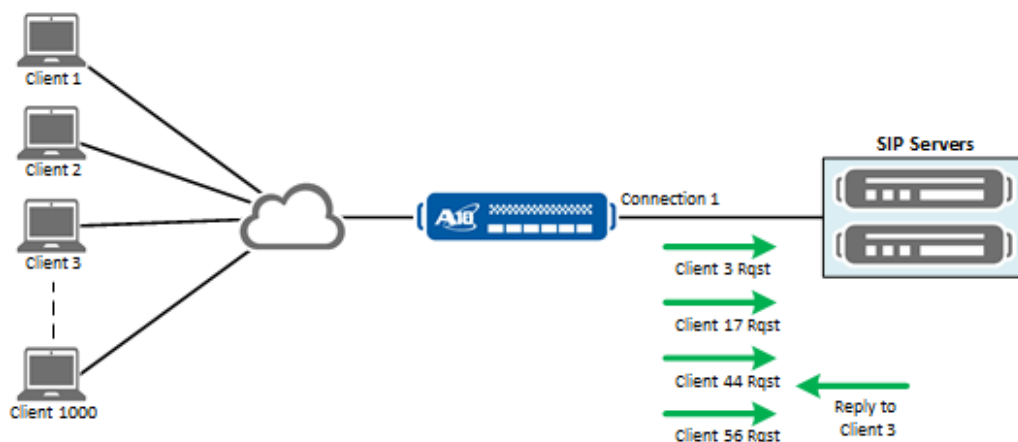


In this example, each SIP server can handle a maximum of 256 client connections. However, there are 1000 SIP clients that use the VIP as their SIP server.

To enable the SIP servers to be used with this many clients, the connection-reuse feature is configured on the ACOS device. The ACOS device is allowed to open a maximum of 100 connections to each server, but uses each connection for multiple clients.

While the ACOS device is sending a client request on a connection, the connection is in use. However, as soon as the request has been sent, the ACOS device frees the connection to be used again. The connection can be used for the same client or another client. The ACOS device does not wait for a reply to the client's request before freeing the connection. [Figure 35](#) shows an example.

Figure 35 : SIP Connection Reuse



In this example, the ACOS device sends requests for 3 different clients before receiving the reply to the first request. To identify the client to which to forward the reply, the ACOS device examines the X-Forwarded-For header in the reply.

The operation of connection reuse for SIP over TCP is different from the operation of connection reuse for HTTP. For HTTP, the ACOS device does not free a connection after sending a client's request. Instead, the ACOS device frees the connection only after receiving a response to the request.

ACOS Requirements

In addition to the requirements for any SIP over TCP / TLS configuration (described in the configuration section), the following items are required for SIP multiplexing:

- Connection-reuse template – Connection-reuse templates have the following options:
 - Timeout – Specifies how long a reusable connection can remain idle before being terminated.
 - Limit per server – Specifies the maximum number of connections to the server. (In [SIP Multiplexing](#), this option would be set to 100)
 - Keep-alive connections – Specifies the number of new reusable connections to open before beginning to reuse the existing connections for new clients.

- Client IP insertion – When this SIP template feature is enabled, the ACOS device inserts an X-Forwarded-For header into the client's request before forwarding the request to the SIP server. The header contains the client's IP address and client port number. ACOS expects the server to send back this header, and uses the header to identify the client to which to send replies from the SIP server.
- Server keepalive (described in [Client Keepalive and Server Keepalive](#)).

Client and Server Requirements

To use the ACOS device as a multiplexer for SIP over TCP/TLS, the clients and SIP servers must meet certain requirements:

- The SIP clients must be able to send SIP pings.
- The SIP server must be able to reply to SIP pings, with SIP pongs.
- The SIP server must be able to include the X-Forward-For header added to the client's request by the ACOS device, in replies to the client.

Client Keepalive and Server Keepalive

ACOS can send and receive SIP keepalive messages:

- Ping – A SIP ping is a 4-byte message containing a double carriage return line feed (CRLF).
- Pong – The reply to a SIP ping is called a “pong”. A pong is a 2-byte message containing a single CRLF.

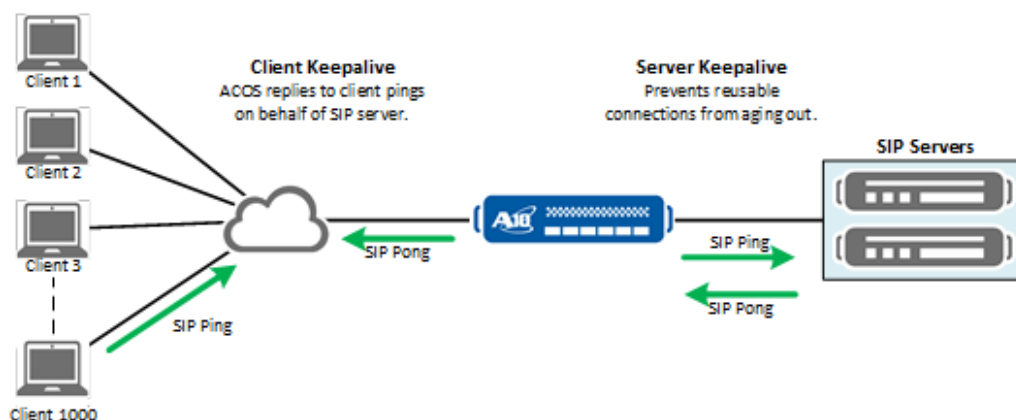
Client keepalive enables the ACOS device to reply to SIP pings sent by clients instead of forwarding them to the SIP server. This feature is applicable regardless of whether you use the ACOS device to multiplex SIP connections.

Server keepalive enables the ACOS device to generate SIP pings and send them to the server. ACOS uses server keepalive to prevent the reusable connections to the server from aging out. If the ACOS device does not receive a pong before the connection-reuse timeout expires, the ACOS device closes the connection. Server keepalives apply only to configurations that include connection reuse, such as a configuration that uses the ACOS device as a SIP multiplexer.

If connection reuse is configured, even with client keepalive disabled, ACOS devices respond to client SIP pings with pongs.

[Figure 36](#) shows an example of a configuration that uses both SIP keepalive features.

Figure 36 : SIP Keepalive



ACOS Actions if Selection of a Client or SIP Server Fails

You can configure the action the ACOS device takes if selection of a SIP client or a SIP server fails. The action can be one of the following:

- Reset the connection. This is the default for client-selection failures and server-selection failures.
- Drop the SIP message.
- Send a message string.
 - Example message sent to client when server can not be reached: “504 Server Time-out”
 - Example message sent to server when client cannot be reached: “480 Temporarily Unavailable”

SLB Network Address Translation for SIP over TCP / TLS

SLB Network Address Translation (NAT) is used for SIP over TCP / TLS load balancing in much the same way SLB NAT is used for load balancing other types of traffic.

When a client sends a SIP request, the request is addressed to the virtual IP address (VIP) and protocol port number configured on the ACOS device for the SIP servers. ACOS translates the destination IP address and port of the request from the VIP to the real IP address and port of a SIP server. ACOS does not change the client IP address or source protocol port number.

Likewise, when the ACOS device receives a SIP packet from a SIP server, the ACOS device translates the source IP address and port from the server's real IP address and SIP port to the VIP address and port, then sends the packet to the client.

By default, the ACOS device also translates the client IP address and protocol port number where they are used in some other parts of the SIP packet. However, the ACOS device *does not* translate server addresses or protocol port numbers in the following headers:

- Call-ID header
- X-Forwarded-For header
- Via headers, except for the top Via header

You can disable translation in any of the following places in the packet:

- Start line
- Individual headers
- Body

For example, if the client is required to be authenticated before registration, and the authentication realm is the VIP instead of a domain name, the ACOS device must not translate the virtual IP address and port into a real server address and port in the Authorization header. Otherwise, authentication will fail.

Configuring SIP over TCP / TLS for SIP Multiplexing

The following sections provide instructions for configuring Load Balancing for SIP over UDP.

- [Configuring SIP over TCP / TLS for SIP Multiplexing \(Procedure\)](#)
- [GUI Configuration](#)
- [CLI Configuration](#)
- [Displaying SIP SLB Statistics](#)

Configuring SIP over TCP / TLS for SIP Multiplexing (Procedure)

To configure an ACOS device for secure SIP multiplexing:

- Optionally, configure a health monitor for SIP over TCP.
- Configure a real server for each SIP server. Use the SIP protocol port number on the server (for example, 5060) as the port number.

Use TCP as the protocol type. Use a Layer 4 TCP health monitor. When you add the TCP port, the default TCP health monitor is automatically applied to the port and enabled.

- Configure a service group containing the real servers.
- Configure a SIP template with at least the following options enabled:
 - Client IP insertion
 - Client keepalive
 - Server keepalive
- Configure a connection-reuse template. Set the limit-per-server option to the maximum number of SIP connections to allow on each SIP server.
- If clients will use SIP over TLS, import the certificates and keys the SIP server would use to authenticate clients. Configure a client-SSL template and add the certificates and keys to the template.
- Configure a virtual server with the IP address to which clients will send SIP requests. For SIP over TLS Clients, add a protocol port with service type “sips”. For SIP over TCP Clients, add a protocol port with service type “sip-tcp”. Bind the port

to the service group, and to the SIP and connection-reuse templates. If a client-SSL template is used, bind the port to the client-SSL template too.

GUI Configuration

The following steps show how to configure the SIP multiplexing as described in [SIP Multiplexing](#).

- [To configure a real server for the SIP multiplexing](#)
- [To configure a service group for the SIP servers and add the servers to the group](#)
- [To configure a SIP template for connection reuse](#)
- [To import a certificate for SSL](#)
- [To add client SSL template](#)
- [To add a virtual server and port](#)

To configure a real server for the SIP multiplexing

1. Navigate to **ADC >> SLB >> Servers** tab.
2. Click **Create**. The Create Server page appears.
3. In the **Name** field, enter a name for the server.
4. Click the **Type** radio button, and then enter the IP address of the server in the field below.
5. In the Port section, click the **Create** button.
6. In the **Port** field, enter “5060” for the SIP port number.
7. Click the **Protocol** drop-down list, and then select **TCP**.
8. Repeat for each server.

To configure a service group for the SIP servers and add the servers to the group

1. Navigate to **ADC >> SLB >> Service Groups** tab.
2. Click **Create**. The Create Service Group page appears.
3. In the **Name** field, enter a name for the service group.
4. In the **Protocol** drop-down list, select **TCP**.

5. In the **Algorithm** drop-down list, select **Round Robin**.
6. In the Member section, click **Create**, and then click the **Server** drop-down list to select the existing server that you just configured.
7. In the **Port** field, enter the SIP port number “5060”.
8. Click **Create**.
9. Repeat for each member.

To configure a SIP template for connection reuse

1. Navigate to **ADC >> Templates >> Application** tab.
2. Click **Create**, and then select **Connection Re-Use** from the drop-down menu that appears.
3. Enter a name for the template in the **Name** field.
4. Configure limits.
5. Click **OK**. The Connection Re-Use template appears in the Application template table.

To import a certificate for SSL

1. Navigate to **ADC >> SSL Management >> SSL Certificates** tab.
2. Click **Import**.
3. Click the **Certificate and Key** radio button for import, then the **Local** radio button to import the certificate from a local system.
4. Choose the certificate/key source file location and click **Import**.

To add client SSL template

1. Navigate to **ADC >> Templates >> SSL** tab.
2. Click **Create >> Client SSL**.
3. In the CA Certs drop-down menu, select the imported certificates and keys.
4. Click **OK**.

To add a virtual server and port

1. Navigate to **ADC >> Templates >> Virtual Servers** tab.
2. Click **Create**.
3. Enter a name in the **Name** field and an IP address in the **IP Address** field.
4. From the **Virtual Port** section, click **Create**.
5. In the protocol drop-down menu, select **SIPS**.
6. In the port field, enter the SIPS port number, "5061".
7. In the **Service Group** drop-down menu, select **service group**.
8. Expand **General Fields**.
9. From the **Source NAT Pool** drop-down menu, select **Source NAT Pool**.
10. From the **ACL** drop-down menu, select your access list and click **Add**.
11. Expand **Templates**.
12. In the **Template Connection Reuse** drop-down menu, select the connection re-use template.
13. In the **Template SIP** drop-down menu, select the SIP template.
14. In the **Template Client SSL** drop-down menu, select the client SSL template.
15. Click **Create**.
16. Click **Update**.

CLI Configuration

This example implements the SIP multiplexing configuration ([SIP Multiplexing](#)).

1. These commands access the configuration level of the CLI and configure a SIP over TCP health monitor:

```
ACOS(config)# health monitor sip-over-tcp
ACOS(config-health:monitor)# method sip tcp
ACOS(config-health:monitor)# exit
```

2. The following commands configure the real servers:

```
ACOS(config)# slb server siptls-rs1 10.4.2.1
```

```

ACOS(config-real server)# port 5060 tcp
ACOS(config-real server-node port)# health-check sip-over-tcp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
ACOS(config)# slb server siptls-rs2 10.4.2.2
ACOS(config-real server)# port 5060 tcp
ACOS(config-real server-node port)# health-check sip-over-tcp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit

```

3. The following commands configure the service group:

```

ACOS(config)# slb service-group siptls-sg tcp
ACOS(config-slb svc group)# member siptls-rs1 5060 10.4.2.1
ACOS(config-slb svc group-member:5060)# exit
ACOS(config-slb svc group)# membersiptls-rs2 5060 10.4.2.2
ACOS(config-slb svc group-member:5060)# exit
ACOS(config-slb svc group)# exit

```

4. The following commands configure the SIP template:

```

(config)# slb template sip siptls-tmplt
ACOS(config-sip)# insert-client-ip
ACOS(config-sip)# client-keep-alive
ACOS(config-sip)# failed-client-selection "480 Temporarily Unavailable"
ACOSACOS(config-sip)# failed-server-selection "504 Server Time-out"
ACOS(config-sip)# exclude-translation header Authentication
ACOS(config-sip)# exit

```

5. The following commands configure the NAT pool and connection-reuse template:

```

ACOS(config)# ip nat pool pool1 10.10.10.100 10.10.10.100 netmask /24
ACOS(config)# slb template connection-reuse siptls-tmplt
ACOS(config-conn reuse)# limit-per-server 100
ACOS(config-conn reuse)# keep-alive-conn 64
ACOS(config-conn reuse)# exit
ACOS(config)# exit

```

6. The following commands import the certificates and keys to use for authenticating SIP clients:

```

ACOS# import cert ca-cert.pem scp:
Address or name of remote host []?192.168.1.1
User name []?admin
Password []?*****
File name [/]?ca-cert.pem
ACOS# import key ca-certkey.pem scp:
Address or name of remote host []?192.168.1.1
User name []?admin
Password []?*****
File name [/]?ca-certkey.pem
ACOS# import cert cert.pem scp:
Address or name of remote host []?192.168.1.1
User name []?admin
Password []?*****
File name [/]?cert.pem
ACOS# import key certkey.pem scp:
Address or name of remote host []?192.168.1.1
User name []?admin
Password []?*****
File name [/]?certkey.pem

```

7. The following commands configure the client-SSL template:

```

ACOS(config)# slb template client-ssl siptls-tmplt
ACOS(config-client ssl)# ca-cert ca-cert.pem
ACOS(config-client ssl)# certificate cert.pem key certkey.pem
ACOS(config-client ssl)# exit

```

8. The following commands configure the virtual server:

```

ACOS(config)# slb virtual-server siptls-vip 10.1.54.4
ACOS(config-slb vserver)# port 5061 sips
ACOS(config-slb vserver-vport)# service-group siptls-sg
ACOS(config-slb vserver-vport)# source-nat pool pool1
ACOS(config-slb vserver-vport)# templatesip siptls-tmplt
ACOS(config-slb vserver-vport)# templateconnection-reuse siptls-tmplt
ACOS(config-slb vserver-vport)# templateclient-ssl siptls-tmplt
ACOS(config-slb vserver-vport)# exit
ACOS(config-slb vserver)# exit

```

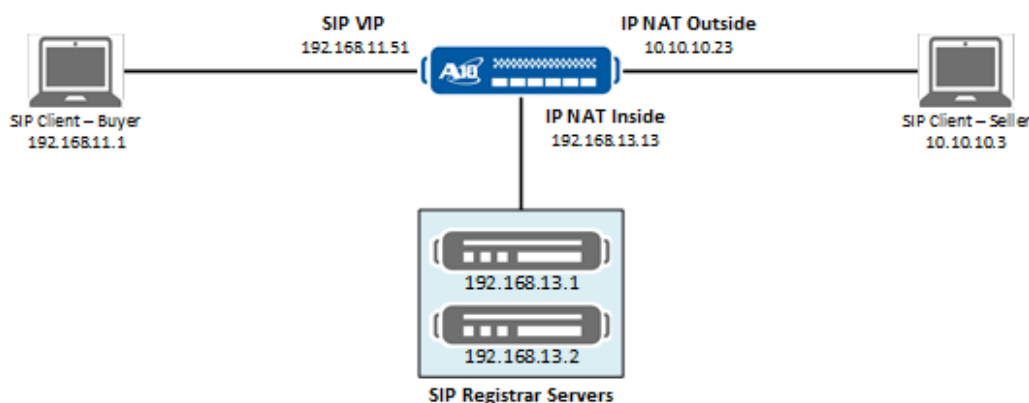
Displaying SIP SLB Statistics

To display SIP SLB statistics, use the `show slb sip` command. Use the `detail` option to display statistics separately for each CPU; otherwise, the command displays aggregate statistics for all CPUs.

Disabling Reverse NAT Based on Destination IP Address

You can use a SIP template to disable reverse NAT for traffic from servers, based on IP address. This option is useful in cases where a SIP server needs to reach another server, and the traffic must pass through the ACOS device. [Figure 37](#) shows an example.

Figure 37 : Reverse NAT Disabled for Traffic from a SIP Server



By default, the ACOS device performs reverse NAT on traffic from a SIP server before forwarding the traffic. Reverse NAT translates the source IP address of return traffic from servers to clients back into the VIP address before forwarding the traffic to clients. If SIP server traffic must reach another server and pass through the ACOS device, the destination server receives the traffic from the VIP address instead of the SIP server address.

GUI Configuration

Configure the extended ACL that matches on the SIP server's subnet and on the database server's subnet.

1. Navigate to **Security >> Access List >> Extended**.
2. Click **Create**.
3. In the ID field, enter an ID for the extended access list.
4. Click **Entry**.
5. In the **Action** drop-down menu, click Permit.
6. In the **Service** field, click IP.
7. For Source Address and Destination Address, click **Subnet**.
8. Enter the Address and Mask for both the Source Address and Destination Address.
9. Click **Create**.

Configure the SIP template to disable reverse NAT for traffic that matches the ACL.

1. Navigate to **ADC >> Templates >> L7 Protocols** tab.
2. Click **Create**, and then select **SIP** from the drop-down menu that appears.
3. Select the **Keep Server IP If Match ACL** checkbox.
4. From the drop-down menu, select the **ACL ID** or **ACL Name** radio button, then select the ID or name.
5. Click **OK** to save your changes.

Bind the SIP template to the SIP virtual port.

1. Navigate to **ADC >> SLB >> Virtual Servers** tab.
2. Click **Create**.
3. Enter the name of the virtual server in the Name field.
4. Enter the IP address of the virtual server in the IP Address field.
5. In **Virtual Port** tab, click **Create**.
6. In the **Protocol** drop-down menu, click **SIP**.
7. In the port field, enter the SIP port number, "5060".
8. Expand **Templates**.
9. In the **Template SIP** drop-down menu, select the created SIP template.

10. Click **Create**.

11. Click **Update**.

CLI Configuration

The commands in this section are applicable to [Figure 37](#).

This command configures an extended ACL that matches the SIP server and database server subnets:

```
ACOS(config)# access-list 101 permit ip 10.10.10.0 /24 10.20.20.0 /24
```

The `keep-real-server-ip-if-match-acl` command configures the SIP template to disable reverse NAT for traffic that matches the ACL:

```
ACOS(config)# slb template sip sip1
ACOS(config-sip)# keep-server-ip-if-match-acl 101
ACOS(config-sip)# exit
```

The following commands bind the SIP template to the SIP virtual port:

```
ACOS(config)# slb virtual-server sip-vip 192.168.11.51
ACOS(config-slb vserver)# port 5060 sip
ACOS(config-slb vserver-vport)# template sip sip1
```

IP NAT for SIP

ACOS supports IP NAT for SIP. This feature allows clients in a private network to make SIP calls to outside SIP servers without revealing client IP addresses to the servers. Dynamic NAT and static NAT are both supported.

Only the SIP signaling packets are NATted. The media packets are not NATted.

To configure IP NAT for SIP:

1. Configure a pool, range list, or static inside source NAT mapping that includes real IP addresses of inside clients.
2. Enable inside NAT on the interface connected to the inside clients.
3. Enable outside NAT on the interface connected to the external SIP servers.

Configuring IP NAT for SIP (CLI Example)

The following configuration excerpt uses dynamic NAT.

```
ACOS(config)# access-list 1 permit any
ACOS(config)# interface ethernet 2
ACOS(config-if:ethernet:2)# ip address 171.1.1.1 255.255.255.0
ACOS(config-if:ethernet:2)# ip nat inside
ACOS(config-if:ethernet:2)# exit
ACOS(config)# interface ethernet 3
ACOS(config-if:ethernet:3)# ip address 2.2.2.1 255.255.255.0
ACOS(config-if:ethernet:3)# ip nat outside
ACOS(config-if:ethernet:3)# exit
ACOS(config)# ip nat pool xin 2.2.2.100 2.2.2.100 netmask /32
ACOS(config)# ip nat inside source list 1 pool xin
ACOS(config)# exit
```

Financial Information eXchange Load Balancing

ACOS supports Financial Information eXchange (FIX) message load balancing. These sections describe how to configure FIX load-balancing and customize a FIX template to direct FIX traffic.

The following topics are covered:

Overview	279
FIX Load Balancing	281
Viewing FIX Traffic Statistics	287

Overview

You can configure FIX load-balancing through the GUI or CLI to direct FIX traffic based on the ID tags of the financial institutions receiving or sending FIX messages. In addition, you can configure the ACOS device to insert the client's IP address into the FIX message header before sending the client request to a FIX server.

Tag-based Service Group Selection

You can configure FIX load balancing to use a default service group and direct a section of client requests to a different service group when the TargetCompID (tag 56) or SenderCompID (tag 49) of the FIX message matches exactly with a specified keyword.

In the following FIX message the TargetCompID is "CUSTOMER6" and the SenderCompID is "FIRM1":

```
8=FIX.4.1 9=112 35=0 49=FIRM1 56=CUSTOMER6 34=240 52=43483994-08:01:54  
112=68981704-08:01:36 10=190
```

An FIX template can direct ACOS to use a different service group for server selection when TargetCompID or SenderCompID of a FIX message matches a keyword. In this example, the keyword "FIRM1" switches service groups based on the SenderCompID, or "CUSTOMER6" to switch service groups based TargetCompID.

Client IP Address Insertion

The Insert Client IP option appends the client's IP address to the FIX message, using the tag 11447.

For example, given the following FIX message:

```
8=FIX.4.1 9=112 35=0 49=FIRM1 56=CUSTOMER6 34=240 52=43483994-08:01:54  
112=68981704-08:01:36 10=190
```

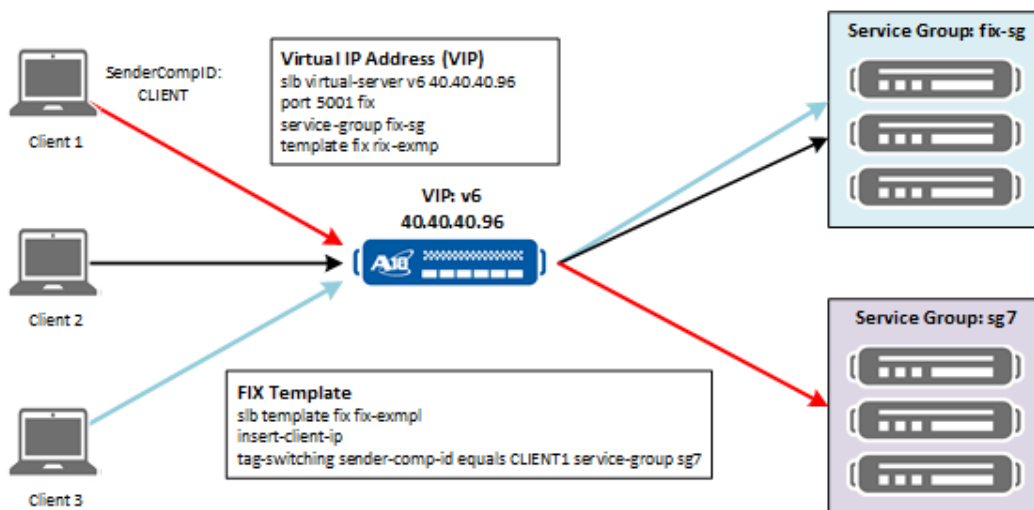
If client IP insertion is enabled in the FIX template and the client's original IP address is 192.168.13.37, then the FIX message header is revised with the 11447 tag, as shown below in bold:

```
8=FIX.4.1 9=112 35=0 49=FIRM1 56=CUSTOMER6 34=240 52=43483994-08:01:54
112=68981704-08:01:36 10=190 11447=192.168.13.37
```

FIX Load Balancing Example

[Figure 38](#) illustrates an example deployment of FIX load balancing. In this example, a FIX template is applied to the virtual IP address (VIP) 40.40.40.96 and is configured for target switching based on the SenderCompID tag. The ACOS device directs a client request to the “sg7” service group if the SenderCompID tag in the FIX message matches the string “CLIENT1”.

Figure 38 : FIX Load-balancing – Deployment Example



The SenderCompID tag in the FIX message from Client 1 equals “CLIENT1” and is sent to the service group “sg7” for server selection. Neither the SenderCompID tags nor TargetCompID tags for Client 2 and Client 3 match a target switching keyword in the FIX template. As a result, the requests from Client 2 and Client 3 are directed to the default “fix-sg” service group, bound to the FIX virtual port.

See [Configuring FIX Load Balancing CLI Example](#) for the configuration procedure of this example.

FIX Load Balancing

This section includes the following FIX Load Balancing Instructions:

- [Configuring FIX Load Balancing](#)
- [GUI Configuration](#)
- [CLI Configuration](#)
- [Configuring FIX Load Balancing CLI Example](#)

Configuring FIX Load Balancing

To configure FIX load balancing:

1. Configure the real servers that will handle FIX traffic and add the TCP port to each server.
2. Configure a service group for the FIX servers and add the servers to the group.
This is the FIX service group.
3. Create a FIX template to redirect all FIX traffic to the FIX service group.
4. Configure a virtual server that contains the FIX port and bind the port to the FIX service group. Add the FIX service group and the FIX template to the port.

GUI Configuration

To configure FIX load balancing from the GUI, perform the following tasks:

Configure Low-Latency

To enable low-latency, Navigate to **ADC >> SLB >> Global >> Low Latency** option.
(Changes will come into effect next time you reload/reboot the Software).

Configure Network Resources

1. Configure Servers:
 - a. .
 - b. Navigate to **ADC >> SLB >> Servers** tab.
 - c. Click **Create**. The Create Server page appears.
 - d. In the **Name** field, enter a name for the server.
 - e. Select the **Type** radio button (IPv4, IPv6, or FQDN) and enter the IP address of the FIX server.
 - f. In the Port section, click the **Create** button.
 - g. In the **Port Number** field, enter a number between 1-65535.
 - h. Click the **Protocol** drop-down menu and select **TCP**.
 - i. For the **Health Check** radio button, select **Monitor**.
 - j. From the **Health Monitor** drop-down menu, select a health monitor to be applied to this port.
 - k. Click **Create**. The port appears in the Port list.
 - l. Click **Create** again. The server appears in the real server table.
 - m. Repeat step a to step l for each real server that will process FIX messages.
2. Configure a Service Group:
 - a. Navigate to **ADC >> SLB >> Service Groups** tab.
 - b. Click **Create**. The Create Service Group page appears.
 - c. In the **Name** field, enter a name for the service group.
 - d. Click the **Protocol** drop-down menu and select **TCP**.
 - e. In the Member section, click the **Create** button to add a server to this service group.
 - f. Select the **Existing Server** radio button.
 - g. Click the **Server** drop-down menu and select the name of a real server from the drop-down menu.
 - h. In the **Port** field, enter the port number.

- i. Click **Create**.
- j. Repeat step a to step j for each server.
- k. Click **Create**. The new service group appears in the service group table.

Configure a FIX Template

1. Navigate to **ADC >> SLB >> L7 Protocols** tab.
2. Click **Create** and from the drop-down menu that appears, select **FIX**. The Create FIX Template appears.
3. Enter a **Name** for the template.
4. Select the Insert Client IP radio button to append the client's IP address to the FIX message header before forwarding the message. (Client IP insertion is disabled by default.)
5. For Tag Switching Type, click the drop-down menu and select one of the following options:
 - Sender Comp ID – The ACOS device selects a service group for FIX requests based on the value of the SenderCompID tag. This tag identifies the financial institution that is sending the request.
 - Target Comp ID – The ACOS device selects a service group for FIX requests based on the value of the TargetCompID tag. This tag identifies the financial institution to which the request is being sent.
6. Click the drop-down menu from the right-most column and select the desired service group. By default, ACOS uses the service group that is bound to the virtual port.
7. If you select the Sender Comp ID or Target Comp ID, these options can be entered in the center field:
 - Equals – Specifies the keyword which the ACOS device will match against the TargetCompID or SenderCompID tag. Enter a 1-63 character string. The keyword is case sensitive and must match exactly with the SendCompID tag or TargetCompID tag. For example, "ABC" is different from "Abc".
 - Service Group – Specifies a service-group to use for client requests that match

the tag value.

8. Click **OK** to save your changes.

Configure a Virtual Server for the FIX Proxy

1. Navigate to **ADC >> SLB >> Virtual Servers** tab.
2. Click **Create**.
3. In the **Name** field, enter a name for the virtual server.
4. Select the **IPv4** or **IPv6** radio button and enter the IP address which will receive clients' FIX messages.
5. In the Virtual Port section, click the **Create** button.
 - Click the **Protocol** drop-down menu and select **FIX**.
 - In the **Port** field, enter the FIX port number. This can be a value between 1-65535.
 - Click **Templates** to expand the template configuration options, and click the **Template FIX** drop-down menu and select the FIX template configured above.
 - Click **Create**. The virtual port appears in the Virtual Port list for the virtual server.
6. Click **Update**. The virtual server appears in the virtual server table.

CLI Configuration

To configure FIX load balancing from the CLI, perform the following tasks:

A complete CLI example is provided in [Configuring FIX Load Balancing CLI Example](#).

Configure Network Resources

1. You can configure the low-latency feature if required, from the CLI, using the following command:

```
ACOS(config)#slb common
ACOS(config-common)#low-latency
```

NOTE: You must reload or reboot the device after configuring this option as it changes the aspects of the system.

2. Configure Servers:

- a. To configure a real server for each FIX server and add the TCP port, use the following commands:
 - Enter the `slb server` command at the global configuration level.
 - Enter the `port` command at the configuration level of the real server.
 - Enter the `health-check` command at the configuration level of the port.
- b. Repeat for each real server that processes FIX messages.

3. Configure a Service Group:

- a. To configure a service group for the FIX servers and add servers to the group:
 - Enter the `slb service-group` command at the global configuration level:
 - Enter the `member` command at the configuration level for the service group:
- b. Repeat for each server.

Configure a FIX Template

1. To create a FIX template, use the `slb template fix` command, which changes the CLI to the configuration level for the template, where the following `insert-client-ip` is available.

This command inserts the client IP address into tag 11447, and recalculates the checksum of the request packet, before sending the request to a FIX server.

2. Optionally, use one of the following commands to configure Tag Switching. If you do not specify this option, ACOS selects the default service group that is bound to the virtual port.

The `tag-switching sender-comp-id equals` command selects a service group for FIX requests based on the SenderCompID tag value. This tag identifies the financial institution that is sending the request.

The **tag-switching target-comp-id equals** command selects a service group for FIX requests based on the TargetCompID tag value, which identifies the financial institution where the request is sent.

Configure a Virtual Server for the FIX Proxy

Use the following commands to configure a virtual server:

1. Use the **slb virtual-server** command at the global configuration level. Specify an IP address to receive FIX traffic. This command changes the CLI to the configuration level for the virtual server.
2. Enter the **port** command to create a port for FIX traffic:
3. Use the **service-group** command to bind the FIX group to the virtual port.
4. Enter the **template fix** command to bind a FIX template to the virtual port.

Configuring FIX Load Balancing CLI Example

This example describes how to configure FIX load balancing and bind a FIX template to a virtual port (vport). These steps are described with respect to the deployment example in [FIX Load Balancing Example](#).

1. Configure the real servers that will handle FIX messages:

```
ACOS(config)# slb server fix-server1 40.40.40.3
ACOS(config-real server)# port 333 tcp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
ACOS(config)# slb server fix-server2 40.40.40.9
ACOS(config-real server)# port 444 tcp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
ACOS(config)# slb server fix-server3 40.40.40.10
ACOS(config-real server)# port 555 tcp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
```

2. Create a service group and add the servers configured above to the group:

```
ACOS(config)# slb service-group fix-sg tcp
ACOS(config-slb svc group)# member fix-server1 333
```

```
ACOS(config-slb svc group-member:333)# exit
ACOS(config-slb svc group)# member fix-server2 444
ACOS(config-slb svc group-member:444)# exit
ACOS(config-slb svc group)# member fix-server3 555
ACOS(config-slb svc group-member:555)# exit
ACOS(config-slb svc group)# exit
```

3. (Not shown) Configure an additional service group to use for FIX messages that contain an ID tag which matches the tag switching keyword:
4. Create a FIX template with tag switching based on the SenderCompID or TargetCompID. In this example, if the client sends a FIX request with tag "49=CLIENT1", the ACOS device uses the alternative service group "sg7" for server selection.

```
ACOS(config)# slb template fix fix-exmpl
ACOS(config-fix)# insert-client-ip
ACOS(config-fix)# tag-switching sender-comp-id equals CLIENT1 service-
group sg7
ACOS(config-fix)# exit
```

5. Create a virtual server and adds the FIX service groups:

```
ACOS(config)# slb virtual-server v6 40.40.40.96
ACOS(config-slb vserver)# port 5001 fix
ACOS(config-slb vserver-vport)# service-group fix-sg
ACOS(config-slb vserver-vport)# template fix fix-exmpl
ACOS(config-slb vserver-vport)# exit
ACOS(config-slb vserver)# exit
```

Viewing FIX Traffic Statistics

This section describes how to view FIX load balancing counters from the ACOS CLI.

To display FIX SLB statistics, use the `show slb fix` command.

```
ACOS(config)# show slb fix
Total
-----
Current proxy conns      4
Total proxy conns        2
Client fail               7
```

Server fail	2
Server selection failure	1
no route failure	0
Source NAT failure	1
Insert client IP	5
Default switching	1
Sender ID switching	4
Target ID switching	0

Server and Port Templates

This section describes how to configure parameters for multiple servers and service ports using server and port templates.

The following topics are covered:

Overview	290
Configuring and Applying Server and Port Templates	292
Port and Server Template Parameters	297

Overview

Server and port templates define a set of parameter values which can be applied to multiple SLB servers and service ports. ACOS supports the following types of templates for configuration of SLB servers and ports:

- Server – Contains configuration parameters for real servers
- Port – Contains configuration parameters for real service ports
- Virtual-server – Contains configuration parameters for virtual servers
- Virtual-port – Contains configuration parameters for virtual service ports

Precedence of Template and Direct Settings

Some template parameters also can also be directly set or changed on an individual server or port.

- When a parameter is set (or changed from its default) in a template and on an individual entity (server or port), the setting on the entity takes precedence over the template setting
- When a parameter is set (or changed from its default) in a template but is not set or changed from its default on an individual entity, the template setting takes precedence over the port and server setting.

Default Server and Port Templates

ACOS has a default template for each of these template types. If you do not explicitly bind a server or port template to a server or port, the default template is automatically applied. For example, when you create a real server, the parameter settings in the default real server template are automatically applied to the new server, unless you bind a different real server template to the server.

The default server and port templates are each named “default”. The default settings in the templates are the same as the default settings for the parameters that can be set in the templates.

If you are upgrading an ACOS device that has a configuration saved under a previous release, the default server and port templates are automatically bound (applied to) the servers and ports in the configuration. This does not change the configuration or operation of the servers and ports themselves, since the default server and port templates use the default settings for all parameters, unless overridden by parameter settings on the individual servers and ports.

The deletion of the SLB template is dependent on the global object deletion configuration. For more information, see *Disabling the Deletion of Referenced Objects* section in the [System Configuration and Administration Guide](#).

Modifying a Default Template

You can modify a default template by creating a new one named “default” and modifying the settings you want to change. The new template replaces the previous one.

In addition to configuring custom server, port, virtual-server, or virtual-port templates, you can modify the default templates. Before changing a default template, make sure the changes you plan to make are applicable to all servers or ports that use the template.

Parameters Configured Through Server and Port Templates

This section includes the parameter summaries.

The following topics are covered:

Real Server Template Parameters	291
Real Port Template Parameters	292
Virtual Server Template Parameters	292
Virtual Port Template Parameters	292

Real Server Template Parameters

To configure a template of SLB settings for real servers, see the `slb template server` command in *Command Line Interface Reference*.

Real Port Template Parameters

To configure a template of SLB settings for ports on real servers, see the `slb template port` command in *Command Line Interface Reference*.

Virtual Server Template Parameters

To configure a template of SLB settings for virtual servers, see the `slb template virtual-server` command in *Command Line Interface Reference*.

Virtual Port Template Parameters

To configure a template of SLB settings for virtual servers, see the `slb template virtual-port` command in *Command Line Interface Reference*.

Configuring and Applying Server and Port Templates

The following topics are covered:

Configuring Server and Port Templates	292
Applying a Server or Port Template	294

Configuring Server and Port Templates

Configure a server and port templates can be configured through the CLI and the GUI.

GUI Configuration

1. Navigate to **ADC >> Templates SLB** tab.
2. Click **Create**, then select one of the following from the drop-down menu to display the configuration page for the specified template:
 - Port
 - Server

- Virtual Port
 - Virtual Server
3. To modify an existing template, either click the template name or the corresponding Edit text.
 4. Enter a name for the template (Only when creating a new template).
 5. Enter or edit the other settings. (See the descriptions in the sections below for information.)
 6. When finished, click **OK** to create a new template for the port, server, virtual port, or virtual server.
 7. Click the **Save** icon at the upper right-most corner of the GUI window.

CLI Configuration

To configure server and port templates, use these commands at the global configuration level.

```
slb template server
slb template port
slb template virtual-server
slb template virtual-port
```

These commands change the CLI to the configuration level for the template. To modify the default template, specify the name “default” (without the quotation marks).

To display the settings in a template, use these commands:

```
show slb template server
show slb template port
show slb template virtual-server
show slb template virtual-port
```

These commands configure a new real server template and bind the template to two real servers. For information about binding templates, see [Applying a Server or Port Template](#).

```
ACOS(config)# slb template server rs-templ1
ACOS(config-rserver)# health-check ping2
ACOS(config-rserver)# conn-limit 500000
ACOS(config-rserver)# exit
```

```

ACOS(config)# slb server rs1 10.1.1.99
ACOS(config-real server)# template server rs-tpmplt1
ACOS(config-real server)# exit
ACOS(config)# slb server rs2 10.1.1.100
ACOS(config-real server)# template server rs-tpmplt1
ACOS(config-real server)# exit

```

Applying a Server or Port Template

When modifying a “default” server or port template, changes automatically apply to servers or ports not bound to a defined template. Server and port templates take effect only if it is bound to a server or port.

[Table 5](#) lists the types of bindings that are supported for server and port templates.

Table 5 : Server and Port Template Bindings

Template Type	Can Be Bound To...
Server	Real servers
Port	Real server ports. You can apply them to real server ports directly or in a service group. Ports within a service group. This method provides more control than binding a template directly to a port. When a template is bound to a port only in a service group (not directly a port), template settings apply to a port only when the port is used by the service group. Settings do not apply to that port when used in other service groups.
Virtual Server	Virtual servers
Virtual Server Port	Virtual server ports

Binding a Server Template to a Real Server

GUI Configuration

1. Navigate to **ADC >> SLB >> Servers** tab.
2. Click the **Edit** link in the Action column for a configured real server.

3. Expand the **Advanced Fields** section.
4. In the Template Server field, select the server template from the drop-down list.
You can also click the **Add** link to create a new server template.
5. Click **Update**.

CLI Configuration

The following example shows how to bind the server template sv_template1 to a real server:

```
ACOS(config)# slb server rs1
ACOS(config-real server)# template server sv_template1
```

Binding a Server Port Template to a Real Server Port

GUI Configuration

1. Navigate to **ADC >> SLB >> Servers** tab.
2. Click the **Edit** link in the Action column for a configured real server.
3. In the **Port** section, click **Edit**.
4. Expand the **Advanced Fields** section.
5. In the Template Port field, select the server port template from the drop-down list.
6. Click **Update**.

CLI Configuration

The following example shows how to bind the port template pt_template1 to a real server port:

```
ACOS(config)# slb server rs1
ACOS(config-real server)# port 80 tcp
ACOS(config-real server-node port)# template port pt_template1
```

Binding a Virtual Server Template to a Virtual Server

GUI Configuration

1. Navigate to **ADC >> SLB >> Virtual Services** tab.
2. Click the **Edit** link in the Action column for a configured virtual server.
3. Expand the **Advanced Fields** section.
4. In the Virtual Server Template field, select the virtual server template from the drop-down list. You can also click the **Add** link to create a new template.
5. Click **Update**.

CLI Configuration

The following example shows how to bind the virtual server template vs_template1 to a virtual server:

```
ACOS(config)# slb virtual-server vs1
ACOS(config-slb vserver)# template virtual-server vs_template1
```

Binding a Virtual Server Port Template to a Virtual Port

GUI Configuration

1. Navigate to **ADC >> SLB >> Virtual Services** tab.
2. Click the **Edit** link in the Action column for a configured virtual server.
3. Expand the **Templates** section at the bottom of the page.
4. In the Template Virtual Port field, select the virtual server port template from the drop-down list.
5. Click **Update**.

CLI Configuration

The following example shows how to bind the default virtual port template to a virtual port:

```
ACOS(config)# slb virtual-server vs1
ACOS(config-slb vserver)# port 80 tcp
```

```
ACOS(config-slb vserver-vport)# template virtual-port default
```

Binding a Server Port Template to a Service Group

GUI Configuration

1. Navigate to **ADC >> SLB >> Service Groups** tab.
2. Click the **Edit** link in the Action column for a configured service group.
3. In the Member pane, select the **Edit** link for the configured real server.
4. In the Template field, select the server port template from the drop-down list.
5. Click **Update**.

CLI Configuration

The following commands bind the server port template `rsp_template` to `rs1` in service group `sg1`:

```
ACOS(config)# slb service-group sg1 tcp
ACOS(config-slb svc group)# member rs1 80
ACOS(config-slb svc group-member:80)# template rsp_template
```

Port and Server Template Parameters

This sections describe parameters that are configurable through port and server templates.

Each section provides information about the specific parameter. For information about accessing the template (GUI panel or CLI configuration mode) and binding the template to a server or port, see [Configuring and Applying Server and Port Templates](#).

The following topics are covered:

Connection Limiting	298
Connection Rate Limiting	300
Server Flap Dampening	301
Slow-Start	302

Request Rate Limiting	305
Graceful Shutdown	306
Gratuitous ARPs for Subnet VIPs	307
TCP Reset Option for Session Mismatch	308
Client Port Preservation	309

Connection Limiting

By default, the ACOS device does not limit the number of concurrent connections on a server or service port. To prevent servers or services from becoming oversaturated, you can set a connection limit. The ACOS device stops sending new connection requests to a server or port when that server or port reaches its maximum allowed number of concurrent connections.

Connection Limit Parameters

Connection limiting can be set in real server templates, real port templates, virtual server templates, and virtual port templates.

To configure connection limits, you can set the following parameters:

- **Connection limit** – Specifies the maximum number of concurrent connections allowed on a server or port. You can specify 0-64000000 (64 million). By default, the connection limit is 64000000 (64 million).
- **Connection resume threshold (real servers or ports only)** – Specifies the maximum number of connections the server or port can have before the ACOS device resumes use of the server or port. You can specify 1-1048575 connections.
- **Reset** – Programs the entity to reset the connection when the connection limit is reached. Available on virtual server and virtual port templates.
- **Logging** – By default, the ACOS device generates log messages when the connection limit is exceeded.

If you change the connection limiting configuration on a virtual port or virtual server with active sessions, or in a virtual-port or virtual-server template bound to the virtual server or virtual port, the current connection counter for the virtual port or server in show command output and in the GUI may become incorrect. Avoid this by

not changing connection limiting configuration when the server or port has active connections.

Setting a Connection Limit

To set a connection limit in a server or port template, use either of the following methods.

GUI Configuration

In the configuration section for the template:

1. Navigate to one of the following:
 - **ADC >> Templates**, click **Create** and select **Server**.
 - **ADC >> Templates**, click **Create** and select **Port**.
2. In the Connection Limit field, enter maximum number of concurrent connections for the server or port.
3. In the Resume field, enter the maximum number of connections the server or port can have before the ACOS device resumes use of the server or port.
4. Click OK.

CLI Configuration

The following commands set the connection limit to 500,000 concurrent connections in a real server template, then bind the template to real servers:

```
ACOS(config)# slb template server rs-tmpl1
ACOS(config-rserver)# conn-limit 500000
ACOS(config-rserver)# exit
ACOS(config)# slb server rs1 10.1.1.99
ACOS(config-real server)# template server rs-tmpl1
ACOS(config-real server)# exit
ACOS(config)# slb server rs2 10.1.1.100
ACOS(config-real server)# template server rs-tmpl1
ACOS(config-real server)# exit
```

Connection Rate Limiting

You can limit the rate at which the ACOS device is allowed to send new connections to servers or service ports. Connection rate limiting is different from slow-start, which temporarily limits the number of new connections per second when TCP/UDP service comes up on a service port. See [Slow-Start](#).

Connection Rate Limiting Parameters

When you configure connection rate limiting, you can set the following parameters:

- **Connection rate limit** – The connection rate limit specifies the maximum of new connections allowed on a server or service port. By default, the connection rate limit is not set.
- **Interval** – The interval specifies whether the connection rate limit applies to one-second intervals or 100-ms intervals. The default is one-second intervals.
- **Action for excess connections (virtual servers or virtual server ports only)** – The action specifies how the ACOS device responds to connection requests after the connection rate has been exceeded. The action can be to silently drop excess connections or to send a reset (RST) to client requesting the connection. The default action is to silently drop the excess connection requests.
- **Logging** – By default, the ACOS device generates log messages when the rate limit is exceeded.

A server or service port that reaches its connection limit is no longer used by the ACOS device.

Configuring Connection Rate Limiting

GUI Configuration

In the configuration section for the template:

1. Select one of the following:

ADC >> Templates, click **Create** and select **Server**.

ADC >> Templates, click **Create** and select **Port**.

2. In the Connection Rate Limit field, enter the desired connection rate limit.
3. In the Per field, select the sampling interval: 100ms or 1 second.
4. Click **OK**.

CLI Configuration

This section displays the method of configuring connection rate limiting in a real server template, then binding the template to real servers.

The commands below configure a connection rate limit of 50000 per 100ms:

```
ACOS(config)# slb template server rs-tmpl1
ACOS(config-rserver)# conn-rate-limit 50000 per 100ms
ACOS(config-rserver)# exit
```

If you configure a limit for a server and also for an individual port, the ACOS device uses the lower limit. For example, if you limit new TCP connections to a real server to 5000 per second and also limit new HTTP connections to 1200 per second, the ACOS device limits connections to TCP port HTTP to 1200 per second.

The commands below bind this template with the configured rate limit to real servers rs1 and rs2:

```
ACOS(config)# slb server rs1 10.1.1.99
ACOS(config-real server)# template server rs-tmpl1
ACOS(config-real server)# exit
ACOS(config)# slb server rs2 10.1.1.100
ACOS(config-real server)# template server rs-tmpl1
ACOS(config-real server)# exit
```

Server Flap Dampening

Server Flap Dampening Description and Parameters

A server flap is defined by consecutive status down-up status reports. Excessive server flapping disrupts traffic flow. Server flap dampening configures a port template to remove the server from the data balancing rotation for a specified period and create a log entry denoting the event.

Server flap dampening is defined by three parameters:

- The number of server flaps that trigger an event.
- The time period for counting flaps before resetting the count.
- The time the server (port) remains out of service after the dampening event is triggered.

Configuring Flap Dampening

Server flap dampening is configured by the dampening-flap command with a slb real port template. The feature is implemented by binding the template to a real port or to a service group that defines a set of ports.

This example configures the template to force a port down if it experiences more than 3 flaps within 20 seconds. The entity remains out of service for 40 seconds.

```
ACOS(config)# slb template port PORT1
ACOS(config-rport)# dampening-flaps 3 flap-period 20 restore-svc-time 40
ACOS(config-rport)#
```

Slow-Start

The slow-start feature allows time for a server or real service port to ramp up after TCP/UDP service on a server is enabled, by temporarily limiting the total concurrent connections on the server or port. You can configure slow-start parameters described in this section in real server templates and real port templates.

The slow-start feature is not used for a port if the real-port template is applied to the port as part of the member configuration in a service group. In this case, if slow-start is configured in the port template, the slow-start settings are ignored for that service-group member.

Ramp-Up Parameters

By default, slow-start allows a maximum of 128 new connections during the first interval (anywhere between 0 and 10 seconds). During each subsequent 10-second interval, the total number of concurrent connections allowed to the server is doubled. Thus, during the first 20 seconds, the server is allowed to have a total of 256 concurrent connections. After 59 seconds, slow-start ends the ramp-up and no longer limits the number of concurrent connections. [Table 6](#) shows the default ramp-up.

Table 6 : Default Slow-Start Ramp-Up

Number of Seconds After Server Restart	Total maximum Concurrent Connections Allowed After Server Restart
0-9	128
10-19	256
20-29	512
30-39	1024
40-49	2048
50-59	4096
60+	Slow-start ends – No limit

The initial ramp-up interval can be any duration from 0 up to the configured interval (10 seconds by default). After the initial ramp up, each subsequent ramp-up occurs at the end of the configured interval.

You can configure the following ramp-up parameters:

- **Starting connection limit** – The starting connection limit is the maximum number of concurrent connections to allow on the server or service port after it first comes up. You can specify from 1-4095 concurrent connections. The default is 128.
- **Connection increment** – Connection increment specifies the amount by which to increase the maximum number of concurrent connections allowed. Use one of these methods to specify the increment:

Scale factor (This is the default.) – The scale factor is the number by which to multiply the starting connection limit. For example, if the scale factor is 2 and the starting connection limit is 128, the ACOS device increases the connection limit to 256 after the first ramp-up interval. The scale factor can be 2-10. The default is 2.

Connection addition – As an alternative to specifying a scale factor, you can instead specify how many more concurrent connections to allow. You can specify 1-4095 new connections.

- **Ramp-up interval** – The ramp-up interval specifies the number of seconds between each increase of the number of concurrent connections allowed. For example, if the ramp-up interval is 10 seconds, the number of concurrent connections to allow

is increased every 10 seconds. The ramp-up interval can be 1-60 seconds. The default is 10 seconds.

- Ending connection limit – The ending connection limit is the maximum number of concurrent connections to allow during the final ramp-up interval. After the final ramp-up interval, the slow start is over and does not limit further connections to the server. You can specify from 1-65535 connections. The default is 4096.

For the connection increment, you can specify a scale factor or a connection addition. The ending connection limit must be higher than the starting connection limit.

If a runtime connection limit is also configured on the server or port (for example, by [Connection Limiting](#)), and the normal connection limit is smaller than the slow-start ending connection limit, the ACOS device limits slow-start connections to the maximum allowed by the normal connection limit.

Behavior When Slow Start Is Also Configured on the Real Server

You can enable slow-start on individual real servers. However, ramp-up settings on individual servers are not configurable. The settings are the same as default ramp-up settings in server and port templates. It is recommended to configure slow start *only* in a server template or port template, not on the real server.

If you do configure slow-start both on the real server itself and in a real server template or real port template, the actual slow-start behavior can differ from the behavior configured in the template.

- If slow start is configured on the real server and in a real server template, the slow-start settings on the real server are used and the settings in the template are ignored. It is recommended to configure slow start only in a real server template or real port template.
- If slow start is configured on the real server and in a real port template, the lower number of connections allowed by either of the configurations at a given interval is used.

Configuring Slow Start

GUI Configuration

In the configuration section for the real server template or real port template:

1. Navigate to one of the following:
ADC >> Templates, click **Create** and select **Server**.
ADC >> Templates, click **Create** and select **Port**.
2. Select the Slow Start checkbox to activate the configuration fields.
3. In the From field, enter the starting connection limit.
4. In the Choose Operator field, select either Add or Times from the drop-down list, then enter the value you want to add or multiply by in the Add or Times field, respectively.
5. Enter the connection increment in the field next to the increment method you selected.
6. In the Every field, enter the desired ramp-up interval.
7. In the Till field, enter the ending connection limit.
8. Click **OK**.

CLI Configuration

To configure slow-start, use the `slow-start` command at the configuration level for a real server or real service port. The following commands enable slow start in a real server template, using the default settings, and bind the template to real servers.

```
ACOS(config)# slb template server rs-tmpl1
ACOS(config-rserver)# slow-start
ACOS(config-rserver)# exit
ACOS(config)# slb server rs1 10.1.1.99
ACOS(config-real server)# template server rs-tmpl1
ACOS(config-real server)# exit
ACOS(config)# slb server rs2 10.1.1.100
ACOS(config-real server)# template server rs-tmpl1
ACOS(config-real server)# exit
```

Request Rate Limiting

You can limit the number of new requests that can be received by a real port. This option applies only to configurations that use an external-service template.

GUI Configuration

On the configuration page for the real port template:

1. Navigate to **ADC >> Templates**.
2. Click **Create** and select **Port** from the drop-down list.
3. In the Request Rate Limit field, enter the number of requests.
4. In the Request Rate Per field, select the sampling interval (per 100ms, or per second).
5. Click **OK**.

CLI Configuration

To configure request rate limiting for real ports, use the `request-rate-limit` command at the configuration level for the real port template.

This example configures a request rate limit of 5000 requests per 100 milliseconds and programs the device to send an RST to a client that sends a new request during an interval when the request rate is exceeded. By default, requests that are received after the limit is exceeded are dropped with no RST.

```
ACOS(config)# slb template port default
ACOS(config-rport)# request-rate-limit 500 per 100ms reset
ACOS(config-rport)# exit
```

Graceful Shutdown

You can configure a grace period for Down servers. ACOS will continue to forward packets to Down ports for the duration of the grace period.

This option is useful for taking servers down for maintenance without immediately impacting existing sessions on the servers. The grace period can be 1-86400 seconds.

- The service group must contain 2 or more servers for this feature to work.
- This feature supports stateless and stateful load balancing. However, the feature is not supported for stateful hash load-balancing methods, such as source-IP-based or destination-IP-based hashing.

GUI Configuration

1. Navigate to **ADC >> Templates**.
2. Click **Create** and select **Port** from the drop-down list.
3. Enter the desired grace period in the Down Grace Period field.
4. Click **OK**.

CLI Configuration

To configure the grace period, use the `down-grace-period` command at the configuration level for the real port template.

```
ACOS(config)# slb template port default
ACOS(config-rport)# down-grace-period 5000
ACOS(config-rport)# exit
```

Gratuitous ARPs for Subnet VIPs

Virtual server templates have an option to enable gratuitous ARPs for all VIPs in a subnet VIP. (A subnet VIP is a range of VIPs created from a range of IP addresses within a subnet.)

By default, the ACOS device sends gratuitous ARPs for only the first IP address in a subnet VIP. You can enable the ACOS device to send gratuitous ARPs for all the IP addresses within a subnet VIP.

This option applies only to VIPs that are created using a range of subnet IP addresses. The option has no effect on VIPs created with a single IP address.

GUI Configuration

1. Navigate to **ADC >> Templates**.
2. Click **Create** and select **Virtual Server** from the drop-down list.
3. Select the **Subnet Gratuitous ARP** check box.
4. Click **OK**.

CLI Configuration

To enable gratuitous ARPs for all VIPs in subnet VIPs, use the **subnet-gratuitous-arp** command at the configuration level for the virtual server template used to configure the VIPs.

The following commands modify the default virtual server template to enable gratuitous ARPs for subnet VIPs. The change applies to all subnet VIPs that use the default template for virtual server configuration.

```
ACOS(config)# slb template virtual-server default
ACOS(config-vserver)# subnet-gratuitous-arp
ACOS(config-vserver)# exit
```

TCP Reset Option for Session Mismatch

Virtual port templates have an option that enables sending of a TCP Reset (RST) in response to a session mismatch. A session mismatch occurs when the ACOS device receives a TCP packet for a TCP session that is not in the active session table on the ACOS device.

This option is useful in cases where a session ages out or is deleted on the ACOS device, but the client does not receive a RST or FIN for the session. In this case, without a RST, the session could remain open on the client until the session ages out. When this option is enabled, TCP RSTs are sent in the cases listed in [Table 7](#).

Table 7 : Processing When Session Is To Be Deleted

Session Termination Method	Packet Type Sent by Client or Server After Session Termination	ACOS Response
Session is terminated by FINs from client and server	Any packet type other than SYN	Maintain connection as long as there is traffic. When there is no traffic, remove the connection one second later.
Session ages out	Any packet type other than SYN	Move session from delete queue back into active session table.

The option is disabled by default, which means the ACOS device does not send a RST in response to a session mismatch. You can enable the option in individual virtual port templates.

This option does not apply to sessions that are in the delete queue. If the ACOS device receives a packet for a session that has been moved to the delete queue, the ACOS device does not send a TCP RST. Instead, the ACOS device reactivates the session and allows it to age out normally.

GUI Configuration

1. Navigate to **ADC >> Templates >> SLB** tab.
2. Click **Create**, then select **Virtual Port** from the drop-down list.
3. Click the checkbox in the Reset Unknown Connection field.
4. Click **Create**.

CLI Configuration

To enable sending of TCP RSTs in response to a session mismatch, use the following command at the configuration level for a virtual port template:

```
ACOS(config)# slb template virtual-port default
ACOS(config-vport)# reset-unknown-conn
```

Client Port Preservation

Virtual-port templates provides 'snat port preserve' feature that attempts to preserve the client's source port for the traffic destined to the virtual port. The port preservation works only when both the source NAT and snat port preserve options are enabled for the virtual port.

Additionally, this feature works on the following two conditions:

- Always attempt snat port preserve: When the configured virtual port matches the configured real port, the show slb l4 respective counters are incremented accordingly. For example,

```
ACOS# show slb l4
```

```
Total
-----
.....
NAT Port Preserve Try 1
NAT Port Preserve Succ 1
.....
```

- **Best Effort:** When the hashing of an incoming client source port and configured real port matches the current CPU and when this condition fails, then neither of the above counters will increment.

Known Limitation

- Port preservation is not always guaranteed and is performed on a best-effort basis.
- Port preservation depends on the number of Platform CPUs. Hence, in some cases, the ports from 1024 - <xxx> will not be preserved.
- Port preservation does not work for FTP active mode sessions.

GUI Configuration

1. Navigate to **ADC >> Templates >> SLB** tab.
2. Click **Create** drop-down list and then select **Virtual Port**.
3. Enter the mandatory fields.
4. Select the **SNAT Port Preserve** checkbox.
5. Click **Create**.

CLI Configuration

The following command configures a NAT pool:

```
ACOS(config)# ip nat pool mypool 10.10.10.40 10.10.10.42 netmask
255.255.255.0
```

These commands configure the virtual-port template and the virtual port.

```
ACOS(config)# slb template virtual-port vport
ACOS(config-vport)# snat-port-preserve
ACOS(config-vport)# exit
```

```
ACOS(config-vport)# slb virtual-server vip1 192.168.25.25  
ACOS(config-slb vserver)# port 80 tcp  
ACOS(config-slb vserver-vport)# source-nat pool mypool  
ACOS(config-slb vserver-vport)# service-group sg1-http  
ACOS(config-slb vserver-vport)# template virtual-port vport
```

Health Monitoring

The following topics are covered:

Overview	313
Health Method Types	319
Configuring and Applying a Health Method	332
Service Group Health Checks	359
In-Band Health Monitoring	364
Consecutive Health Checks	367
Maintenance Health Checks	368
On-Demand Health Checks	370
Enabling Strict Retries	370
Globally Changing Health Monitor Parameters	371
Health-Check Rate Limiting	373
Multi-CPU Support for Health Monitoring	376
Database Health Monitors	377
Kerberos Health Monitors	378
Compound Health Monitors	380
Configurable Response Codes: RADIUS Health Monitors	385
Displaying Health Status	385
Using External Health Methods	389

Overview

ACOS devices can regularly check the health of real servers and service ports. Health checks ensure that client requests go only to available servers. Servers or ports that respond appropriately to health checks remain eligible to serve client requests. A server or port that does not respond appropriately to a health check is temporarily removed from service, until the server or port is healthy again.

You can configure health methods on the ACOS device by configuring settings for the type of service you are monitoring. You also can configure health monitors externally using scripts and import the monitors for use by the ACOS device.

For information about health monitoring of the ACOS device's nexthop gateways, see the "Gateway Health Monitoring" section in the *System Configuration and Administration Guide*.

The following topics are covered:

Default Health Checks	313
Health-Check Guidelines	314
Health Monitor Natting	315
Health Method Timers	317
Health Check Operations	318

Default Health Checks

ACOS performs the following types of health checks by default:

- Layer 3 ping – Every 5 seconds, the ACOS device sends an ICMP echo request (ping) addressed to the real server's IP address. The server passes the health check if it sends an echo reply to the ACOS device. If the server does not reply after the fourth attempt (the first attempt followed by 3 retries), the ACOS device sets the server state to DOWN.
- Layer 4 TCP – Every 5 seconds, the ACOS device sends a connection request (TCP SYN) to the specified TCP port on the server. The port passes a health check if it

replies to the ACOS device by sending TCP SYN ACK. If the port does not reply after four attempts, the ACOS device sets the port state to DOWN.

- Layer 4 UDP – Every 5 seconds, the ACOS device sends a packet with a valid UDP header and a garbage payload to the UDP port. The port passes the health check if it either does not reply, or replies with any type of packet *except* an ICMP Error message. If the port replies with an ICMP Error message, the ACOS device sets the port state to DOWN.

The default ICMP, TCP, or UDP monitor is not used if you disable it on the server or port, or you apply a different monitor to the server or port.

- In the GUI, on the server configuration page, the default health monitors appear as “(default)” in the Health Monitor drop-down lists for the server itself and for the individual TCP or UDP ports.
- For the server itself, “(default)” means the Layer 3 ping described above. For TCP ports, “(default)” means the Layer 4 TCP health monitor described above. Likewise, for UDP ports, “(default)” means the Layer 4 UDP health monitor described above.
- On new ACOS devices, the Health Monitor list contains one health monitor (ping), which is the same as the “default” server health monitor. The list does not contain default TCP or UDP health monitors.
- The health monitor ping configuration runs at the global level and affects all partitions.

Health-Check Guidelines

By default, Layer 3 health checking of real server IP addresses is enabled. Likewise, Layer 4 (TCP and UDP) health checking of server ports is enabled by default.

Layer 3 health checking can be CPU-intensive. To help reduce unnecessary processing overhead, if you are using Layer 4 or Layer 7 health checking of a real server’s ports, it is recommended to disable Layer 3 health checking of that server’s IP address.

For large deployments (1K Services), A10 recommends disabling default Layer 3 health check and using only Layer 4-7 health checks. (See [Globally Disabling Layer 3 Health Checks](#).)

Health Monitor Natting

In a VRRP-A deployment, both the active and standby devices send health check traffic to the back-end server. The Firewall has to maintain rules for each interface IP that sends health check traffic.

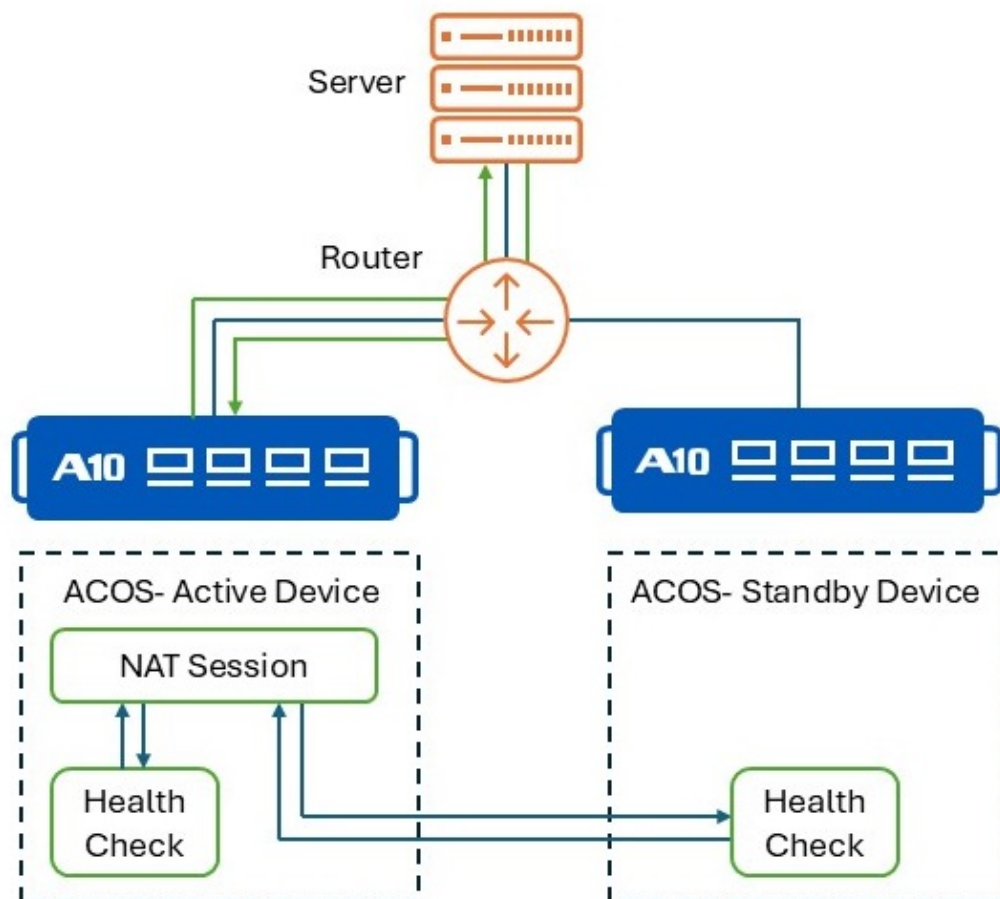
ACOS provides a way to consolidate the source IP used for health check traffic, where only one IP from Smart NAT/ NAT pool is used for all health checks. This reduces the number of firewall rules and simplifies traffic management.

In VRRP-A, the "enable-vrrp-a-mode" command ensures that ACOS forwards health check packets only to the real server when the device is active. In standby mode, the device routes health checks to the active device.

When the active device receives health check traffic from health check module or standby device, it creates a NAT session, translates the source IP, and forwards the traffic to the real server. This ensures that only the active device forward health check traffic to the server. When this feature is enabled, health check only uses the NAT IP.

[Figure 39](#) shows an example.

Figure 39 : Health Monitor Natting in VRRP-A Deployment



CLI Configuration

To configure the source NAT/smart NAT in health check, use the following commands:

```
ACOS(config)#health source-nat
ACOS(config-health:source-nat)#source-nat-pool <pool-name>
ACOS(config-health:source-nat)#source-nat-pool-v6 <pool-name-v6>
ACOS(config-health:source-nat)#smart-nat-precedence[vrid <vrid-value>]
ACOS(config-health:source-nat)#enable-vrrp-a-mode interface
<ethernet|ve|trunk> <intf-num>
!
```


Limitations

- This feature requires a dedicated interface for VRRP-A communication.
- Health check traffic from a standby device may be identified as SLB traffic when directed to a wild card VIP (Virtual IP). A precisely defined access list must be configured to prevent this problem.
- Health checks may fail for a short period of time during failover due to the time required to establish a NAT session on the new active device.
- Health monitor objects cannot be referenced across partitions.

Health Method Timers

Health methods operate based on the following timers:

- Interval – Number of seconds between health check attempts. Determination of server or port health is not made during the interval. Instead, health determination is made after the server or port passes or fails one attempt (interval), or the number of retries is exhausted.

Default interval is 5 seconds. You can change it to a value from 1-180 seconds.

- Timeout – Number of seconds the ACOS device waits for a reply to a health check. If the ACOS device does not receive the expected reply by the end of the timeout, the ACOS device either sends the health check again (if there are retries left) or marks the server or service down. You can specify 1-180 seconds. The default is 5 seconds.

The type of reply expected by the ACOS device depends on the monitor type. (See [Health Method Types](#))

- Retries – Maximum number of times the ACOS device sends the same health check to an unresponsive server or service before marking that server or service as down. You can specify 1-10 attempts. The default is 3.
- Up-Retry – Number of consecutive times the device must pass the same periodic health check, in order to be marked as Up. You can specify 1-63 attempts. The default is 1. (See [Consecutive Health Checks](#)).

The default interval and timeout can be adjusted automatically by health-check rate limiting. (See [Health-Check Rate Limiting](#)). The timeout does not apply to externally configured health monitors.

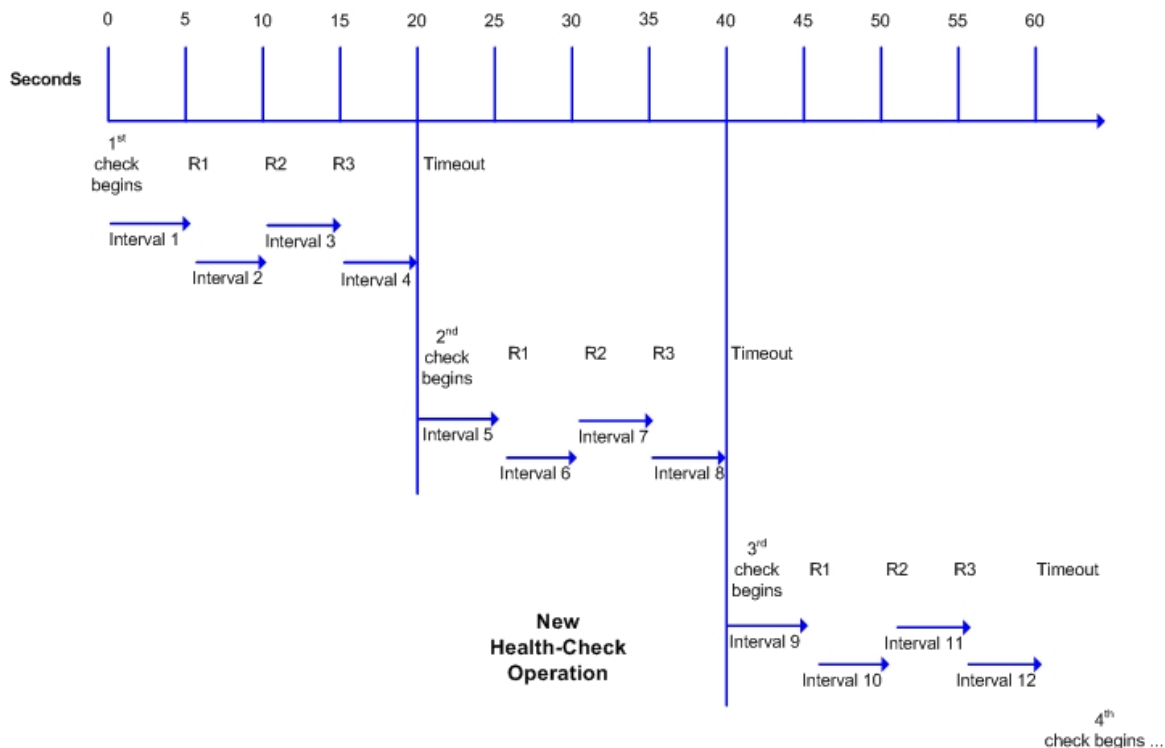
Health Check Operations

Server or Port Is Unresponsive

[Figure 40](#) shows how health checking operates when the server or port is unresponsive.

After each interval, ACOS immediately begins the next health check, because the next interval begins immediately after the previous attempt times out. In the figures, “R” indicates a retry.

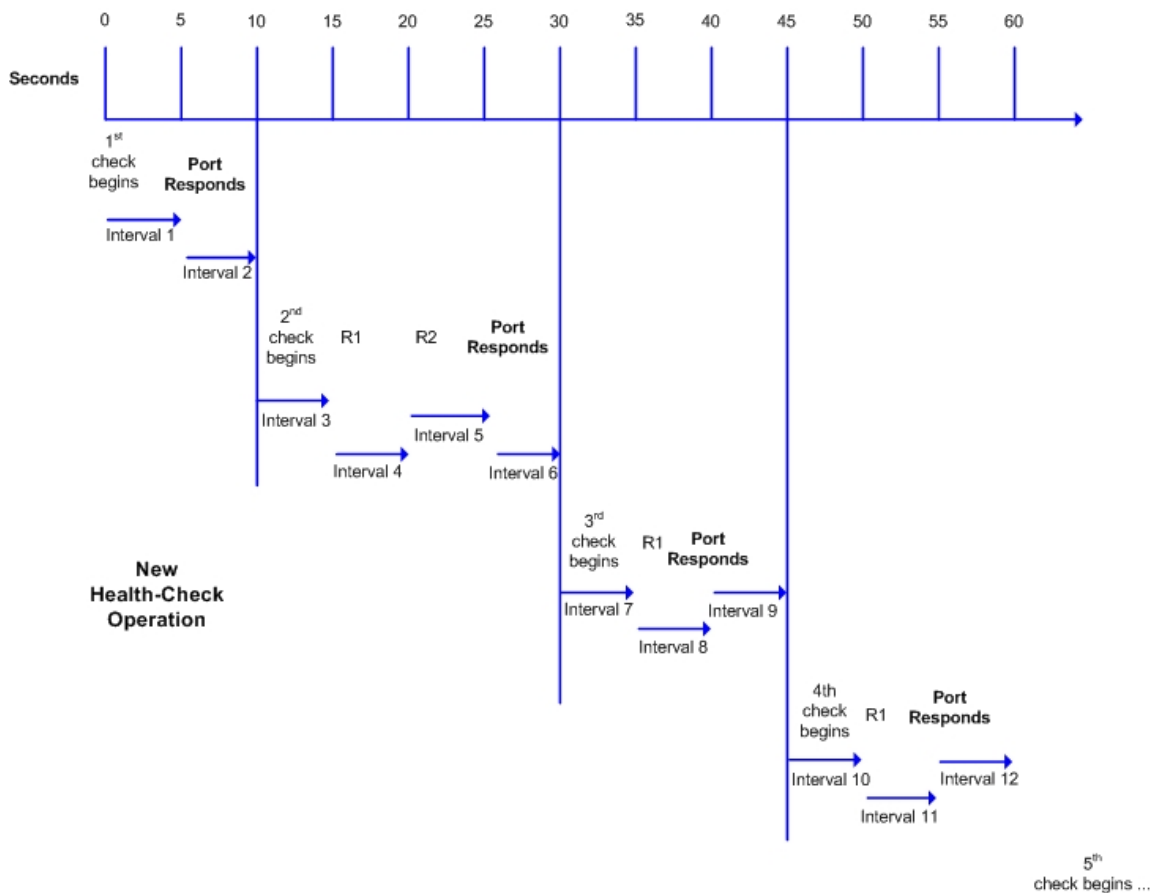
Figure 40 : Health Checks Using Default Settings – Server or Port Is Unresponsive



Server or Port Is Responsive

[Figure 41](#) shows health checking operation when the server or port is responsive. ACOS begins the next health check when the next interval begins which, using the default interval value, the next interval begins within 5 seconds.

Figure 41 : Health Checks Using Default Settings – Server or Port Responds



Health Method Types

[Table 8](#) lists the internal health method types supported by the ACOS device. The health methods use the well-known port numbers for each application by default. You can change the port numbers and other options when you define the health methods.

Multiple health method instances can be defined using the same method type and different parameters. Likewise, multiple health monitors can use the same health method to check different servers.

When a health monitor is in use by a server, the monitor cannot be removed.

To configure a health monitor for Direct Server Return (DSR), see [Health Monitoring Virtual IP Addresses in DSR Deployments](#).

Table 8 : Internal Health Method Types

Type	Description	Successful If...	Configuration Required on Target Server
Database	ACOS device sends a query to the database. The database can be one of the following types: • Oracle • MSSQL • MySQL • PostgreSQL	Server sends a reply with the expected query results. For replies that consist of multiple results, the results are in a table. You can specify the row and column location within the results table to use as the receive string. (For more information, see Database Health Monitors.)	Database must contain the queried data. Requested user name and password must be valid on the server.
DNS	ACOS device sends a lookup request for the specified domain name or server IP address. By default, recursion is allowed. The tested DNS server is allowed	Server sends a reply with the expected status code (0 by default) and record type (A	Domain name in the lookup request must be in the server's database.

Table 8 : Internal Health Method Types

Type	Description	Successful If...	Configuration Required on Target Server
	to send the health check's request to another DNS server if the tested server can not fulfill the request using its own database. Optionally, you can disable recursion.	by default). You can configure the response code(s) and record type required for a successful health check. You can require the server to reply with specific status codes within the range 0-15. For health checks sent to a domain name, you can require the server to reply with one of the following record types: • A – IPv4 address record (the default) • CNAME – Canonical name record for a DNS alias	

Table 8 : Internal Health Method Types

Type	Description	Successful If...	Configuration Required on Target Server
		• SNI - Server Name Indication record by which the client indicates which hostname is attempting to connect, and the real server returns a security certificate . • SOA – Start of authority record • PTR – Pointer record for a domain name • MX – Mail Exchanger record • TXT – Text string • AAAA – IPv6 address record (For more	

Table 8 : Internal Health Method Types

Type	Description	Successful If...	Configuration Required on Target Server
		information, see Customizing DNS Health Monitors)	
FTP	ACOS device sends an FTP login request to the specified port. If anonymous login is not used, the username also must be specified in the health check configuration.	Server replies with FTP OK message or Password message. If the server sends the Password message, the ACOS device sends the password specified in the health check configuration. In this case, the ACOS device expects the server to reply with another OK message.	Requested user name and password must be valid on the server.
HTTP / HTTPS	ACOS device sends HTTP GET, HEAD, or POST request to the specified TCP port and URL. • GET requests the entire page. • HEAD requests only meta-information in the header.	Server replies with OK message (200), by default. You can configure the response code(s) and record type required for a	Requested page (URL) must be present on the server. For GET requests, the

Table 8 : Internal Health Method Types

Type	Description	Successful If...	Configuration Required on Target Server
	POST attempts to write information to the server. For POST requests, you must specify the target field names and the values to post. (For more information, see POST Requests in HTTP/HTTPS Health Monitors) If a user name and password are required to access the page, they also must be specified in the health check configuration. By default, the real server's IP address is placed in the request header's Host: field. You can configure a different value. The following types of authentication are supported: basic and digest authentication. If you specify a username and password, the health monitor will try to use basic authentication first. If this try succeeds, the authentication process is complete. Otherwise, the health monitor will negotiate with the server to select another authentication method, and retry the health check using that authentication method. For HTTPS health checks, the	successful health check. For GET requests, the server also must reply with the requested content or meta-information in the page header. The response must include the string specified in the Expect field on the ACOS device. For HEAD requests, the ACOS device ignores the Expect field and only checks for the server reply message. For POST operations, the data must be posted without error.	string specified as the expected reply must be present. For POST operations, the field names specified in the health check must be present on the requested page. For HTTPS health checks, SSL support must be enabled on the server. A certificate does not need to be installed on the ACOS device. ACOS always accepts the

Table 8 : Internal Health Method Types

Type	Description	Successful If...	Configuration Required on Target Server
	ACOS device encapsulates SSLv3, TLSv1, TLSv1.1, or TLSv1.2 hello messages within the SSLv2 hello messages by default. You can disable this encapsulation. Proxy header v1 and v2 are supported.		server certificate presented by the server.
ICMP	ACOS device sends an ICMP echo request (ping) to the server. The transparent option is applicable for testing the path from an ACOS device to a target device through an intermediate device. This is a Layer 3 health check only. Use the other method types to check the health of a specific application.	Server replies with an ICMP echo reply message.	Server must be configured to reply to ICMP echo requests.
IMAP	ACOS device sends an IMAP user login request with the specified user parameter.	Server replies with an OK message. The ACOS device then sends the password specified in the health check configuration. The ACOS device expects the server to reply	Requested user name and password must be valid on the server.

Table 8 : Internal Health Method Types

Type	Description	Successful If...	Configuration Required on Target Server
		with another OK message.	
Kerberos	Depending on the method options used, a Kerberos health monitor can check accessibility to each of the following: • Key Distribution Center (KDC) Ticket Granting Ticket (TGT) service – Tests ability to access the Kerberos server as a new client requesting a TGT. • Kerberos administrative system – Tests ability to access the Kerberos server as a user account administrator. • Kerberos password change system – Tests ability to access the Kerberos server as a user attempting to change their password.	Kerberos server responds and allows access.	Valid administrator and end-user accounts.
LDAP	ACOS sends an LDAP request to the LDAP port. Optionally, the request can be directed to a specific Distinguished Name. SSL can be enabled for the health check. The ACOS also must send a valid password, if one is required by	Server sends a reply containing result code 0.	If a Distinguished Name and password are sent in the health check, they must match these values on the LDAP server.

Table 8 : Internal Health Method Types

Type	Description	Successful If...	Configuration Required on Target Server
	the server.		A certificate does not need to be installed on the ACOS device. The ACOS device always accepts the server certificate presented by the server.
NTP	ACOS device sends an NTP client message to UDP port 123.	Server sends a standard NTP 48-byte reply packet.	NTP service must be running.
POP3	ACOS device sends a POP3 user login request with the specified user parameter.	Server replies with an OK message. The ACOS device then sends the password specified in the health check configuration. The ACOS device expects the server to reply with another OK message.	Requested user name and password must be valid on the server.

Table 8 : Internal Health Method Types

Type	Description	Successful If...	Configuration Required on Target Server
RADIUS	ACOS device sends a Password Authentication Protocol (PAP) request to authenticate the user name and password specified in the health check configuration.	Server sends Access-Accepted message (reply code 2). You can specify response code(s) that ACOS accepts as valid responses. For example you can configure a health monitor to accept an Access-Reject response (code 3) as a valid response from a healthy RADIUS server.	Requested user name and password must be configured in the server's user database. Likewise, the shared secret sent in the health check must be valid on the server.
SIP	ACOS device sends a SIP OPTION request or REGISTER request.	Server replies with 200 - OK.	None.
SMTP	ACOS device sends an SMTP Hello message.	Server sends an OK message (reply code 250).	Server recognizes and accepts the domain of sender. If SMTP service is running and can reply to Hello messages, the server

Table 8 : Internal Health Method Types

Type	Description	Successful If...	Configuration Required on Target Server
			can pass the health check.
SNMP	ACOS device sends an SNMP Get or Get Next request to the specified OID, from the specified community.	Server replies with the value of the OID.	Requested OID and the SNMP community must both be valid on the server.
TACPLUS	ACOS device configures a method to check server availability by passing the TACACS+ parameters with secret and password encryption.	Server returns a pass message that keeps the server status marked as up.	Requested user name and password must be configured in the server's user database. Also, the shared secret sent in the health check must be valid on the server.
TCP	ACOS device sends a connection request (TCP SYN) to the specified TCP port on the server. Proxy header v1 and v2 are supported.	Server replies with a TCP SYN ACK. By default, the ACOS device completes the TCP handshake	Destination TCP port of the health check must be valid on the server.

Table 8 : Internal Health Method Types

Type	Description	Successful If...	Configuration Required on Target Server
		with the server: ACOS -> Server SYN -> <- SYN-ACK ACK -> FIN -> <- FIN-ACK ACK -> To configure the ACOS device to send a RST (Reset) instead of sending the first ACK, enable the Halfopen option. In this case, the health check is performed as follows: SYN -> <- SYN-ACK RST ->	
UDP	ACOS device sends a packet with a valid UDP header and a garbage	Server does either of the	Destination UDP port of

Table 8 : Internal Health Method Types

Type	Description	Successful If...	Configuration Required on Target Server
	payload to the specified UDP port on the server.	following: - Replies from the specified UDP port with any type of packet. - Does not reply at all. The server fails the health check only if the server replies with an ICMP Error message.	the health check must be valid on the server.

Protocol Port Numbers Tested by Health Checks

If you specify the protocol port number to test in a health monitor, the protocol port number configured in the health monitor is used if you send an on-demand health check to a server without specifying the protocol port. (See [On-Demand Health Checks](#).)

After you bind the health monitor to a real server port, health checks using the monitor are addressed to the real server port number instead of the port number specified in the health monitor's configuration. In this case, you can override the IP address or port using the override options described in [Overriding the Target IP Address or Protocol Port Number](#).

Configuring and Applying a Health Method

1. Create a new health monitor and configure its settings for the type of service you are monitoring and the ciphers for the monitor to use. If you created the monitor externally with a script, import the script.

The GUI does not support selecting the SSL ciphers ACOS uses. To select SSL ciphers for the health monitor feature, use the CLI.

2. Apply the monitor to the real server (for Layer 3 checks) or service port.

You can apply a health monitor to a server or port in either of the following ways:

- Apply the health monitor to a server or port template, then bind the template to the server or port.
- Apply the health monitor directly to the individual server or port.

Configuring an Internal Monitor

Configure an Internal Monitor (GUI Example)

This example creates TCP a health monitor called “hm1:”

1. Navigate to **ADC >> Health Monitors**.
2. Click **Create**.
3. In the **Name** field under the General Fields section, enter **hm1**.
4. Click the **Method type** drop-down menu, and select TCP.

The rest of the configuration fields change depending on which type of health monitor you select. (See [Health Method Types](#))

5. Click **Create Monitor**. The new monitor appears in the Health Monitor table.

Configure an Internal Monitor (CLI Example)

The following example configures an internal TCP health monitor named “hm1:”

```
ACOS(config)# health monitor hm1
ACOS(config-health:monitor)# method tcp port 80
```


The *method-name* can be one of the types listed in [Health Method Types](#). Also see that section for additional options you can specify. For syntax information, see the “Config Commands: SLB Health Monitors” section in the *Command Line Interface Reference*.

Import an Externally Configured Monitor

Import an Externally Configured Monitor (GUI Example)

This example imports an external monitor called “ext-hm:”

1. Create a script for the monitor. (For an example, see [Using External Health Methods](#).)
2. In the GUI, select **ADC > Health Monitors**.
3. From the menu bar, select the **External Programs** tab.
4. Click **Create**.
5. In the **Name** and **Description** fields, enter a name and description for the external health method.
6. Copy and paste the script into the **Definition** field.
7. Click **Create**. The method appears in the External Program Name table.

Import an Externally Configured Monitor (CLI Example)

Create an external Tcl script for the monitor. (For an example, see [Using External Health Methods](#).)

The following example imports the external health monitor named “ext-hm” using ftp:

```
ACOS(config)# import health-external ext-hm ftp://examplehost/script/ext-hm
```

After the external monitor script is imported, create a new health monitor named “hm2” to use the script:

```
ACOS(config)# health monitor hm2
ACOS(config-health:monitor)# method external program ext-hm
```

Apply a Health Monitor to a Real Server or Real Service Template

Apply a Health Monitor to a Real Server Template (GUI Example)

This example applies the health monitor “hm1” to a new real server template named “rs-template.”

1. Navigate to **ADC >> Templates >> SLB** tab.
2. Create a real server template, and apply the health monitor to the template:
 - a. Click **Create** to create the template, and select **Server** from the drop-down list.
 - b. In the Name field, enter **rs-template**.
 - c. In the Health Monitor field, select **hm1** from the drop-down list or available health monitors.
 - d. Click **OK**.

Apply a Health Monitor to a Real Service Port Template (GUI Example)

This example applies the health monitor “hm1” to a new real service port template named “rsp-template.”

1. Select **ADC >> Templates >> SLB** tab.
2. Create a real server port template, and apply the health monitor to the template:
 - a. Click **Create** to create the template, and select **Port** from the drop-down list.
 - b. In the Name field, enter **rsp-template**.
 - c. In the Health Monitor field, select **hm1** from the drop-down list or available health monitors.
 - d. Click **OK**.

Apply a Health Monitor to a Real Server Template or Real Service Port Template (CLI Example)

The following example applies the health monitor “hm1” to real server template “svr-template:”

```
ACOS(config) # slb template server svr-template
```

```
ACOS(config-rserver) # health-check hml
```

Apply a Health Monitor to a Real Server or Service Port

GUI Configuration

Layer 3 health checking can be CPU-intensive. To help reduce unnecessary processing overhead, if you are using Layer 4 or Layer 7 health checking of a real server's ports, it is recommended to disable Layer 3 health checking of that server's IP address.

1. Navigate to **ADC >> SLB >> Servers**.
2. Select the server or click **+ Create** to create a new one.
3. To apply a Layer 3 health monitor to the server, select the health monitor from the Health Monitor drop-down list.
4. To apply a health monitor to a service port:
 - a. In the Port section, click **Create**.
 - b. In the Port Number field, enter the port number.
 - c. Click the **Protocol** drop-down menu and select **TCP** or **UDP**.
 - d. Next to the Health Check, select the desired radio button from these choices: Default, Disable, Monitor, or Follow Port
 - e. If you select the Monitor radio button, the **Health Monitor** drop-down menu appears.
 - f. Click the menu and select the desired health monitor from the list.
5. Enter or change other settings if needed, such as Connection Limit or No Logging.
6. Click **+ Create**.
7. Apply a Health Monitor to a service group. Navigate to **ADC >> SLB >> Service Groups**.
8. Select one of the existing service groups that appear, or click **+ Create** to add a new one.
9. Click the **Health Monitor** drop-down list and select the desired health monitor.

10. Enter or change other settings if needed.

11. Click **Create** or **Update**.

For more information about how health monitors are used when applied to service groups, see [Service Group Health Checks](#).

CLI Configuration

The following example applies the health monitor “hm1” to a service port on real server “rs1:”

```
ACOS(config)# slb server rs1
ACOS(config-real server)# port 80 tcp
ACOS(config-real server-node port)# health-check hm1
```

Health Monitoring Virtual IP Addresses in DSR Deployments

Layer 3 and Layer 4-7 health checks are supported in DSR configurations.

The target of the Layer 3 health checks can be the real IP addresses of the servers, or the virtual IP address, depending on your preference.

- To send the Layer 3 health checks to the real server IP addresses, you can use the default Layer 3 health method (ICMP).
- To send the Layer 3 health checks to the virtual IP address instead:
 - Configure an ICMP health method with the transparent option enabled, and with the alias address set to the virtual IP address.
 - Globally enable DSR health checking.

Layer 4-7 health checks are sent to the same IP address as the Layer 3 health checks, and then addressed to the specific protocol port. You can use the default TCP and UDP health monitors or configure new health monitors. This example uses the default TCP health monitor.

NOTE:

- In DSR route mode, we support configuring the health monitor at the service group configuration level. If you configure the health monitor at the real server, real server port, or service group member configuration level, it may not work in some network topology. For example, when there is one or more routers between ACOS and real server, the traffic may not send to the real server.
- In DSR deployment, external health monitoring is not currently supported.

Configuring a Layer 3 health method targeted to a Virtual IP Address

GUI Configuration

1. Navigate to **ADC >> Health Monitors**.
2. Click **Create**. The Create Health Monitors page appears.
3. In the **Name** field, enter a name for the monitor.
4. Click the **Method Type** drop-down list and select **ICMP**. The ICMP section appears in the pane in the right.
 - a. Select the **Transparent** checkbox in the ICMP section below.
 - b. Select the desired address type, **IPv4** or **IPv6**, and enter the loopback address.
5. Click **Create Monitor**.

CLI Configuration

Use the following commands to configure a health method target to VIP 192.168.4.4:

```
ACOS(config)# health monitor hml  
ACOS(config-health:monitor)# method icmp transparent 192.168.4.4
```

Globally enable DSR health checking of virtual IP addresses

GUI Configuration

1. Navigate to **ADC >> SLB >> Global** tab.
2. From the menu bar, select the
3. Select the **DSR Health Check Enable** checkbox.
4. Click **Update**.

CLI Configuration

Use the following commands:

```
ACOS(config)# slb common
ACOS(config-common)# dsr-health-check-enable
```

(Enter the **slb common** command from the global configuration level to access the configuration level for system-wide SLB parameters.)

Using Send and Receive Strings in TCP Health Checks

ACOS supports use of send and receive text strings in TCP health checks. Send and receive string allow you to add additional intelligence to TCP health monitors. ACOS sends the specified send string to the target TCP port, and watches for the specified reply.

For example, you can configure a send string that is an HTTP GET request, and specify the response string the server must send in reply to the GET request. (See the CLI example below.)

TCP health monitors that include send and receive strings work as follows:

1. ACOS performs the TCP three-way handshake with the TCP port on the server:

```
ACOS          Server
SYN ->
              <- SYN-ACK
ACK ->
```

2. If the three-way handshake is successful, ACOS sends the entire send string to

the TCP port.

```
ACOS          Server
ACK ->
```

3. If the port's reply contains the specified receive string anywhere within the reply, the port passes the health check.

```
ACOS          Server
              <- ACK
```

4. ACOS and server complete the handshake.

```
ACOS          Server
FIN ->
              <- FIN-ACK
ACK ->
```

Configuring Send and Receive Strings in TCP Health Checks (GUI Example)

The following example configures a TCP health monitor named “tcp-with-http-get” that sends an HTTP GET request to TCP port 80, and expects the string “200” to be present in the reply:

1. Navigate to **ADC >> Health Monitors**.
2. Click **Create**. The Create Health Monitors page appears.
3. In the **Name** field, enter `tcp-with-http-get`.
4. Click the **Method Type** drop-down list and select **TCP**. The TCP section appears.

In the Send field, enter the following:

```
GET / HTTP/1.1\r\nHost: 22.1.2.2\r\nUser-Agent: a10\r\nAccept: */*\r\n\r\n
```

In the Response Contains field, enter 200.

5. Click **Create Monitor**.

Configuring Send and Receive Strings in TCP Health Checks (GUI Example)

The following example configures a TCP health monitor named “tcp-with-http-get” that sends an HTTP GET request to TCP port 80, and expects the string “200” to be present in the reply:

```
ACOS(config)# health monitor tcp-with-http-get
ACOS(config-health:monitor)# method tcp port 80 send "GET /
HTTP/1.1\r\nHost: 22.1.2.2\r\nUser-Agent: a10\r\nAccept: */*\r\n\r\n"
response contains 200
```

This health monitor sends an HTTP GET request to TCP port 80 on the target server. This particular request uses the following header fields:

- Host – Specifies the host (server) to which the request is being sent.
- User-Agent – Identifies the entity (user agent) that is sending the request. In this example, the sending entity is “a10”.
- Accept – Specifies the types of media that are allowed in the response. This example uses wildcards (*/) to indicate that any valid media type and range are acceptable.

If the string “200” is present anywhere in the reply from the port, the port passes the health check.

Using Customized Header Insertion in SIP Health Check

ACOS supports sending the SIP requests with the hostname in the request URI instead of IP address. This functionality is beneficial in the environments where the integrated health monitor SIP or SBC server requires a SIP hostname rather than an IP address. You can also configure the customized header content, which will append the SIP health check request header after the original header. The existing headers are not overwritten.

A maximum of eight lines of header-insert fields can be configured, and the header content format should be *name:value* pair. If a header with the same name already exists in the request, the new *name:value* pair is added after the current *name:value* pair.

NOTE: The header insertion feature is supported only for SIP method for both SIP over UDP and SIP over TCP/TLS.

CLI Configuration

1. Configure the SIP health monitor.

```
ACOS(config)#health monitor sip_monitor
ACOS(config-health:monitor)# method sip hostname a10networks.com
ACOS(config-health:monitor)# header-insert Expires:1800
ACOS(config-health:monitor)# header-insert Reply-
To:Bob<sip:bob@biloxi.com>
ACOS(config-health:monitor)# exit
```

NOTE: Space is also supported in the header content with a double quotation. For example, header-insert "Expires: 1800".

2. Bind the health monitor to the real server port.

```
ACOS(config)# slb server sv 10.10.20.11
ACOS(config-real server)# health-check-disable
ACOS(config-real server)# port 5060 udp
ACOS(config-real server-node port)# health-check sip_monitor
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
```

Show Command

Use the **show health monitor** command to view the SIP health monitor status or use **show health stats** to view the detailed send/Failed/Retry/Timeout statistics.

```
ACOS# show health monitor SIP
Monitor Name:          SIP
Interval:              5
Max Retry:             3
Timeout:               5
Up-Retry:              1
SSL-Ciphers:           DEFAULT
Header-Insert:         Expires:1800
Header-Insert:         Reply-To:Bob<sip:bob@biloxi.com>
Passive Enable
Status:                In Use
Method:                SIP
Attribute:              option
```

```
hostname=a10networks.com
```

```
port=5060
```

```
transport-protocol=udp
```

```
Service information:
```

Service	IP address	Port	Status	Reason (Up/Down)

sv	10.10.20.11	5060	UP	SIP Receive OK

Certificate and Key Support in HTTPS Health Monitors

You can add an SSL certificate and key to an HTTPS health monitor. When you use this option, the ACOS device uses the certificate and key during the SSL handshake with the HTTPS port on the server.

The certificate you plan to use with the health monitor must be present on the ACOS device before you configure the health monitor.

CLI Configuration

To add a certificate and key to an HTTPS health monitor, use the `cert` and `key` options with the `method https` command.

The following commands configure an HTTPS health monitor that uses a certificate:

```
ACOS(config)# health monitor https-with-key
ACOS(config-health:monitor)# method https cert cert-for-health-checks
key key1
```

GUI Configuration

1. Navigate to **ADC >> Health Monitors**.
2. Click **+ Create**.
The Create Health Monitors page appears.
3. In the **Name** field, enter the health monitor name.
4. Click the **Method Type** drop-down list and select **HTTPS**.

The HTTPS section appears.

- a. In the Client certificate field, select the certificate from the drop-down list.
 - b. In the Client Private Key field, select the key from the drop-down list.
 - c. If applicable, enter the pass phrase in the Key Password Phrase field.
5. Click **Create Monitor**.

SNI Support for HTTPS Health Monitors

You can add a Server Name Indication (SNI) check for HTTPS health monitoring to add granularity to the health check.

SNI specifies the hostname to client connection. This allows a server to present multiple certificates on the same IP address and TCP port number and multiple secure (HTTPS) websites that are served by the same IP address without requiring all those sites to use the same certificate. Without SNI, each hostname requires its own IP address, to install the SSL certificate. Virtual hosting requires this type of health monitor.

CLI Configuration

To add SNI support to an HTTPS health monitor, use the `sni` and `host` option with the `method https` command. The following commands configure an HTTPS health monitor that uses a certificate:

```
ACOS(config)# health monitor https-mon
ACOS(config-health:monitor)# method https sni host www.example.com
```

To configure the certificate name with SNI, use the following command:

```
ACOS(config-health:monitor)# method https host a10networks.com expect-
cert-name aa sni
```

The configured health monitor can be viewed as part of `show running-config health monitor` command.

```
ACOS1(config)#show running-config health monitor
health monitor https-mon
    method https host www.test.com sni
!
health monitor example.com
    method https host www.example.com sni
```

```
!  
health monitor https-mon  
  method https host a10networks.com expect-cert-name aa sni  
!  
health monitor example.com  
  method https host www.example.com sni
```

GUI Configuration

1. Navigate to **ADC >> Health Monitors**.
2. Click **Create**. The Create Health Monitors page appears.
3. In the **Name** field, enter the health monitor name.
4. Click the **Method Type** drop-down list and select **HTTPS**. The HTTPS section appears.

Enter the user-defined SNI, for example, `www.test.com` in the **Host** field in HTTPS selection.

5. Click **Create Monitor**.

TLS Session Resumption Support for HTTPS Health Monitors

You can configure Session Ticket TLS extension for HTTPS health monitoring when doing SSL handshake.

The TLS session resumption allows the TLS server to restart sessions while avoiding the storage of per-client session state. The session state is encapsulated into a ticket by the TLS server and forwarded to the client. The client will then use the received ticket to reopen a session. The session ticket mechanism is also referred to as the stateless resumption mechanism. For information about this mechanism, refer to [RFC 5077](#).

When HTTPS monitor uses specific SSL cipher suites to match the specified SSL ciphers used by the resource pool, the full SSL handshake negotiation may impact the CPU utilization and performance. On the other hand, the TLS session resumption reuses the recently valid TLS session ticket and resumes using a single resumption

request rather than a full handshake negotiation. This efficiently manages large volumes of HTTPS traffic and ensures the fastest performance.

NOTE: The SSL session ticket feature only applies to the old method HTTPS on the control plane.

CLI Configuration

The following commands configure SSL session ticket for HTTPS health monitor:

For more detailed information, refer *Command Line Reference for ADC guide*.

- To configure SSL session ticket:

```
ACOS(config)# health monitor hm-https
ACOS(config-health:monitor)# ssl-ticket
ACOS(config-health:monitor)# method https
```

- To configure SSL session ticket timeout value:

```
ACOS(config)# health monitor hm-https
ACOS(config-health:monitor)# ssl-ticket lifetime 4200
ACOS(config-health:monitor)# method https
```

- To configure preferred SSL version for negotiation:

```
ACOS(config)# health monitor hm-https
ACOS(config-health:monitor)# ssl-version 34 34
ACOS(config-health:monitor)# method https
```

- To automatically clear all the SSL session ticket:

```
ACOS(config)# clear health https ssl-tickets
```

- To manually clear the specified health monitor SSL session ticket:

```
ACOS(config)# clear health https ssl-tickets hm-https
```

GUI Configuration

To enable the SSL session resumption feature, do the following:

1. Navigate to **ADC >> Health Monitors**.
2. Click **Create**. The Create Health Monitors page appears.

3. In the **General Fields** section, enter the following information:

- SSL Ticket
- SSL Ticket Lifetime(Seconds)
- SSL Version
- SSL Downgradable Version

4. Click **Create Monitor**.

To view the HTTPS statistics, do the following:

- Navigate to **ADC >> Health Monitor >> Statistics >> HTTPS**.

To clear the specific health monitor SSL session ticket, do the following:

1. Navigate to **ADC >> Health Monitor >> Statistics >> HTTPS**.
2. Click **Clear SSL Ticket**.
3. On the **Clear SSL Ticket** dialog, select the health monitor and click **OK**.

POST Requests in HTTP/HTTPS Health Monitors

You can specify a POST operation in an HTTP or HTTPS health monitor. A POST operation attempts to post data into fields on the requested page.

GUI Configuration

The following example configures an HTTP health method names “http1” that uses a POST operation to post `firstname=abc` and `lastname=xyz` to `/postdata.asp` on the tested server.

1. Navigate to **ADC >> Health Monitors**.
2. Click **Create**. The Create Health Monitors page appears.
3. In the Name field, enter `http1`.
4. In the Method Type section, select HTTP from the drop-down list. Configuration fields for HTTP health monitoring options appear.
5. Configure an HTTP POST operation:

- a. Select the checkbox in the URL Type field.
- b. In the URL Type field, select **POST** from the drop-down list.
- c. In the Post Path field, enter `/postdata.asp`.
- d. In the Post Type field, select **POST Data**.
- e. In the Post Data field, enter `firstname=fname&lastname=lname`.

Use “=” between a field name and the value you are posting to it. If you post to multiple fields, use “&” between the fields. The string can be up to 255 bytes long.

6. Click **Create Monitor**.

CLI Configuration

The following commands configure an HTTP health method names “http1” that uses a POST operation to post `firstname=abc` and `lastname=xyz` to `/postdata.asp` on the tested server.

```
ACOS(config)# health monitor http1
ACOS(config-health:monitor)# method http url POST /postdata.asp postdata
firstname=fname&lastname=lname
```

In the `postdatastring`, use “=” between a field name and the value you are posting to it. If you post to multiple fields, use “&” between the fields. For example:

`postdata fieldname1=value&fieldname2=value`. The string can be up to 255 bytes long.

To use POST data longer than 255 bytes, you must import a POST data file and use the `POST / postfile filename` option. To import POST data file up to 2 Kbytes long, use the following command at the global configuration level of the CLI:

The following commands import a file containing a large HTTP POST data payload (up to 2 Kbytes), and add the payload to an HTTP health monitor:

```
ACOS(config)# import health-postfile long-post
scp://exampleuser@192.168.1.1/scripts/
...
ACOS(config)# health monitor http1
ACOS(config-health:monitor)# method http url post / postfile long-post
expect def
```

In this example, health checks that use this health monitor will send a POST request containing the data in “postfile”, and expect the string “def” in response.

Automatic Interval Adjust Based on HTTP Status Code

Passive health monitoring optimizes health monitoring by increasing the health-check interval when the servers are Up.

Passive health monitoring checks the HTTP status code in the initial server reply to a client request. If the server sends enough replies that contain a status code indicating normal operational status, ACOS increases the health-check interval for that server. By increasing the health-check interval, ACOS reduces network overhead.

If a server’s healthy response codes fall below a configured threshold, the health monitor’s configured interval is used again, to more frequently check server health.

Passive Health-Monitoring Parameters

Passive health monitoring is activated and deactivated based on the following parameters. You can configure these parameters in individual HTTP health monitors.

This feature has the following configurable parameters:

- Healthy status code numbers – The set of status codes that indicate the HTTP service is healthy. You can specify one of the following:
 - Any 2xx status code
 - Any status code *other than* a 5xx code

You must specify the set of status codes to use when you configure the monitor. There is no default.

- Passive interval – The health-monitor interval that is used when passive health monitoring is activated. For proper operation of the feature, the passive interval should be longer than the health monitor’s interval. You can specify 1-180 seconds. The default is 10 seconds.
- Sample threshold – Minimum number of server replies that must contain one of the specified status codes, within a given one-second interval, before passive health monitoring is enabled. The sample threshold helps prevent passive health

monitoring from taking effect after only a small total number of samples are taken. You can specify 1-10000 samples per second. The default is 50.

- **Threshold** – Minimum percentage of server replies that must contain a healthy status code, within a given one-second interval, before passive health monitoring is activated. You can specify 0-100 percent. The default is 75 percent. If you specify 0, this parameter is disabled, in which case there is no minimum threshold.

Passive Health-Monitoring Activation

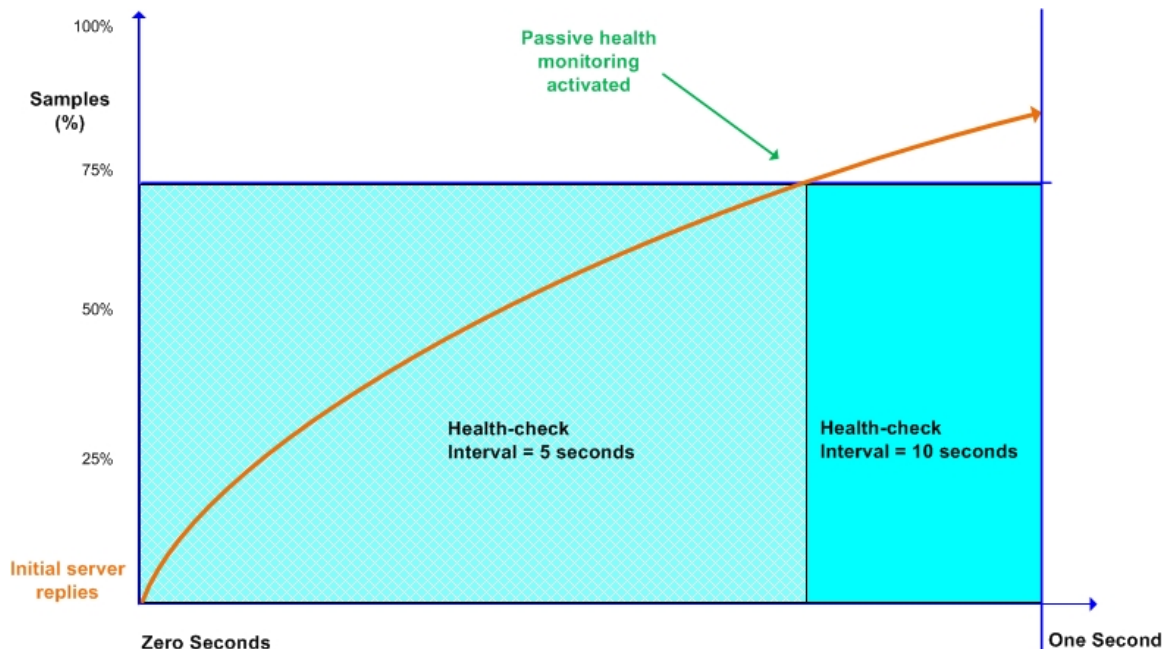
Once per second, these parameters evaluate server responses to initial client requests. Within that interval:

- If the threshold is exceeded after the sample threshold is exceeded, passive interval becomes active.
- If sample threshold and threshold are not exceeded, health monitor interval setting remains in effect.

Response statistics are reevaluated each second. Generally, once a server is up and healthy, the passive interval remains in effect for that server. Health-check traffic overhead for the server is reduced by half.

[Figure 42](#) shows an example of how passive health monitoring is activated.

Figure 42 : Activation for passive health monitoring



In this example, the health monitor interval and each of the passive health-monitoring parameters described above are left at their default values.

When the server first comes up, the health-check interval is set to 5 seconds, which is the interval setting on the health monitor. The server's responses quickly exceed the sample threshold and threshold, so passive health-monitoring mode is activated. This results in the health-check interval being reset to 10 seconds.

The longer interval remains in effect as long as the server responses exceed the thresholds.

- This feature applies only to TCP virtual ports, and only to the ICMP and HTTP health methods.
- Only the first packet in the reverse direction of a TCP session flow is inspected.
- If connection-reuse is enabled on the virtual port, only the response packet for the initial session is examined.

Configuring Passive Health-Monitoring Activation

GUI Configuration

1. Navigate to **ADC >> Health Monitors**.
2. Click **Create** or click the **Edit** link next to the name of a configured ICMP or HTTP health monitor.
3. If configuring a new monitor, enter a name and select the Method Type, ICMP or HTTP.
4. Select the Passive checkbox.
5. Customize any other settings as desired.

For information about the options, see [Passive Health-Monitoring Parameters](#).

6. If configuring an HTTP health monitor, configure HTTP settings as applicable to your deployment.
7. Click **Create Monitor** or **Update Monitor**.
8. Apply the monitor to the server or server port.

CLI Configuration

To configure inband health monitoring based on HTTP status code, use the **passive** command at the HTTP health monitor configuration level. See [Passive Health-Monitoring Parameters](#).

The following commands create a new health monitor, and enable passive health-monitoring mode:

```
ACOS(config)# health monitor http-passive  
ACOS(config-health:monitor)# passive status-code-2xx
```

The following command sets the method to HTTP:

```
ACOS(config-health:monitor)# method http  
ACOS(config-health:monitor)# exit
```

The following commands configure a real server, service group, and virtual server. The HTTP health monitor configured above is applied to the TCP port on the real server.

```
ACOS(config)# slb server ser1 172.168.1.107  
ACOS(config-real server)# health-check-disable  
ACOS(config-real server)# port 80 tcp  
ACOS(config-real server-node port)# health-check http-passive  
ACOS(config-real server-node port)# exit  
ACOS(config-real server)# exit  
ACOS(config)# slb service-group sg1 tcp  
ACOS(config-slb svc group)# member ser1 80  
ACOS(config-slb svc group-member:80)# exit  
ACOS(config-slb svc group)# exit  
ACOS(config)# slb virtual-server vs1 172.168.6.100  
ACOS(config-slb vserver)# port 80 tcp  
ACOS(config-slb vserver-vport)# service-group sg1  
ACOS(config-slb vserver-vport)# exit  
ACOS(config-slb vserver)# exit
```

Customizing DNS Health Monitors

ACOS provides the following configurable options for DNS health monitors:

- Expected:
 - response codes – You can specify a list of response codes, in the range 0-15, that are valid responses to a health check. If the tested DNS server responds with any of the expected response codes, the server passes the health check. By default, the expect list is empty, in which case the ACOS device expects status code 0 (No error condition).
 - ipv4 address - You can specify the IPv4 address expect. This expect can only be specified with type A.
 - ipv6 address - You can specify the IPv6 address expect. This expect can only be specified with type AAAA.
 - FQDN - You can specify the fully qualified domain name. This expect can only be specified with type CNAME, PTR and MX.

The FQDN length can be between 1-255. Only alphabet, digit, hyphen, and period are allowed. Each label first and last character must be digit or letter, max length of each label is 63.

NOTE: When you configure ipv4 address, ipv6 address, and FQDN for the domain name and the IP address then the built-in health check for DNS is enabled. Else, you may need to use external health-check shell script.

- Recursion setting (enabled or disabled) – Recursion specifies whether the tested DNS server is allowed to send the health check's request to another DNS server if the tested server can not fulfill the request using its own database. Recursion is enabled by default.

Record type expected from the server – You can specify one of the following record types:

- A – IPv4 address record
- CNAME – Canonical name record for a DNS alias
- SOA – Start of authority record
- PTR – Pointer record for a domain name
- MX – Mail Exchanger record

- TXT – Text string
- AAAA – IPv6 address record

By default, the ACOS device expects the DNS server to respond to the health check with an A record.

CLI Configuration

To configure a DNS health monitor, use the `health monitor` and `method dns` commands.

- The following commands configure a DNS health monitor that sends a query for `www.example.com`, and expects an Address record and any of the following response codes in reply: 0, 1, 2, 3, or 5. It then binds the monitor to a server port.

```
ACOS(config)# health monitor dnshml
ACOS(config-health:monitor)# method dns domain www.example.com expect
response-code 0-3,5
ACOS(config)# slb server ser1 172.168.1.109
ACOS(config-real server)# port 53 udp
ACOS(config-real server-node port)# health-check dnshml
```

- The following commands configure a DNS health monitor that expect `ipv4-addr` as an answer for type A resolution from a server:

```
ACOS(config)# health monitor hml
ACOS(config-health:monitor)# method dns domain a10networks.com expect
ipv4-addr 10.2.1.12
```

- The following commands configure a DNS health monitor that expect FQDN as an answer for IPv6 reverse resolution from a server:

```
ACOS(config)# health monitor hml
ACOS(config-health:monitor)# method dns ipaddress <ipv6 address> expect
FQDN <fqdn>
```

GUI Configuration

1. Navigate to **ADC >> Health Monitors**.
2. Click **Create**.
3. In the General Fields section, enter a name for the monitor in the Name field.

4. In the Method Type section, select DNS from the drop-down list. Configuration fields for DNS health monitoring options appear.
5. If the DNS server to be tested does not listen for DNS traffic on the default DNS port (53), edit the port number in the Override Port field.
6. To test a specific server, enter the address in the IPv4 Address field. Otherwise, to test based on a domain name sent in the health check, select Domain and enter the domain name in the Domain field.
7. If you select Domain, select the record type the server is expected to send in reply to health checks. Select the record type from the Domain Type drop-down list.
8. If you do not want to allow recursion, select **Disabled** in the IPv4 Recurse drop-down menu.
9. To specify the response codes that are valid for passing a health check, enter the codes in the Domain Response Code field. To specify a range, use a dash. Separate the codes (and code ranges) with commas.
10. Click **Create Monitor**.
11. Apply the monitor to the server or server port.

DNS Health Monitoring over TCP

ACOS also supports use of DNS health monitoring over TCP.

GUI Configuration

On the **DNS health monitor** configuration page, select the IPv4, IPv6, or Domain TCP option, depending on which DNS type you select.

CLI Configuration

To enable TCP instead of UDP for a DNS health monitor, use the `tcp` option with the `method dns` command.

The following example configures a health monitor for DNS over TCP:

```
ACOS(config)# health monitor dns-tcp
ACOS(config-health:monitor)# method dns domain www.example.com tcp
ACOS(config)# slb server ser1 172.168.1.110
```

```
ACOS(config-real server)# port 53 tcp  
ACOS(config-real server-node port)# health-check dns-tcp
```

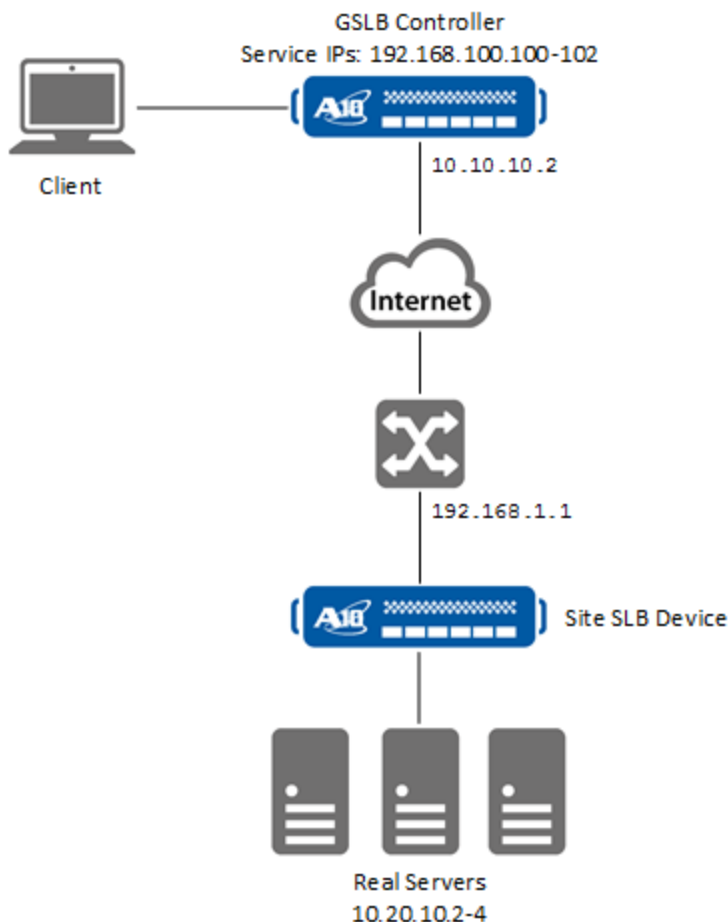
Overriding the Target IP Address or Protocol Port Number

ACOS provides options to override the real server IP address or protocol port number to which health checks are addressed.

By default, the ACOS device sends a Layer 3 health check to the IP address used in the real server configuration. Likewise, the ACOS device sends Layer 4-7 health checks to the real port number in the real server's configuration. For GSLB service IPs, the ACOS device sends the health check to the service IP address. For example, if the configuration has a Layer 3 health monitor used by service IPs 192.168.100.100-102, the ACOS device addresses the health checks those IP addresses.

You can specify an override IP address or protocol port number in the health monitor. In this case, the ACOS device always addresses the health check to the override address or port. This option is particularly useful for testing the health of a remote link, as in the following example.

Figure 43 : Example of Health-check Address Override



In this example, the real servers managed by the site ACOS are configured as service IPs 192.168.100.100-102 on the GSLB ACOS. The health-check metric is enabled in the GSLB policy, so health checks are needed to verify that the service IPs are healthy. One way to do so is to check the health of the ISP link connected to the site ACOS device.

Because the GSLB ACOS device is deployed in route mode instead of transparent mode, the transparent option for ICMP health monitors can not be used to check the remote end of the path. In this case, the health monitor can be configured with an override IP address, 192.168.1.1, to check the health of the ISP link to the site where the servers are located. When the ACOS device in this example uses the health monitor to check the health of a service IP, the device addresses the health check to

192.168.1.1, the override address, instead of addressing the health check to the service IP addresses.

Override Parameters

You can independently configure any of the following override parameters for a health monitor:

- Target IPv4 address
- Target IPv6 address
- Target Layer 4 protocol port number

The override is used only if applicable to the method (health check type) and the target. An IP address override is applicable only if the target has the same address type (IPv4 or IPv6) as the override address.

A protocol port override is applicable to all health methods except ICMP. If the protocol port number is explicitly configured for the method, the override port number is still used instead.

Configuring the Target IP Address or Protocol Port Number Override

CLI Configuration

Use the `override-ipv4`, `override-ipv6`, or `override-port` commands at the configuration level for the health monitor.

The following example configures a health monitor for the service IPs shown in [Example of Health-check Address Override](#), and apply the monitor to the service IPs.

```
ACOS(config)# health monitor site1-hm
ACOS(config-health:monitor)# method icmp
ACOS(config-health:monitor)# override-ipv4 192.168.1.1
ACOS(config-health:monitor)# exit
ACOS(config)# gslb service-ip gslb-srvcl 192.168.100.100
ACOS(config-service-ip:gslb-srvcl)# health-check site1-hm
ACOS(config-service-ip:gslb-srvcl)# exit
ACOS(config)# gslb service-ip gslb-srvc2 192.168.100.101
ACOS(config-service-ip:gslb-srvc2)# health-check site1-hm
ACOS(config-service-ip:gslb-srvc2)# exit
ACOS(config)# gslb service-ip gslb-srvc3 192.168.100.102
```

```
ACOS(config-service-ip:gslb-srv3)# health-check site1-hm  
ACOS(config-service-ip:gslb-srv3)# exit
```

GUI Configuration

1. Navigate to **ADC >> Health Monitors**.
2. Click edit next to the health monitor name or click **Create** to create a new one.
3. For an ICMP health monitor:
 - a. Leave ICMP selected in the Method Type drop-down list.
 - b. Enter the target IP address of the health monitor, in the Override IPv4 or Override IPv6 field.
4. For other health methods, select the type, then enter the target protocol port number in the Override Port field.
5. Click **Create Monitor** or **Update Monitor**. The health monitor list re-appears.

Basing a Port's Health Status on Another Port's Health Status

You can configure the ACOS device to base a real port's health status on the health status of another port number. The real port and the port to use for the real port's health status must be the same type, TCP or UDP.

GUI Configuration

1. Navigate to **ADC >> SLB >> Servers >> *server_name* >> Port >> Create**.
2. Select **Follow Port** in the Health Check field.
 - a. Select **TCP** or **UDP** from the drop-down list in the Follow Port Protocol field.
 - b. In the Follow Port field, enter the port number where you want to base the health of the real port.
3. Click **Create**.

CLI Configuration

Use the `health-check-follow-port` command at the configuration level for the real port.

The following example configures the real port (80) to follow TCP port 8081:

```

ACOS(config)# slb server rs1 10.10.10.10
ACOS(config-real server)# port 8081 tcp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# port 80 tcp
ACOS(config-real server-node port)# health-check-follow-port 8081 tcp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit

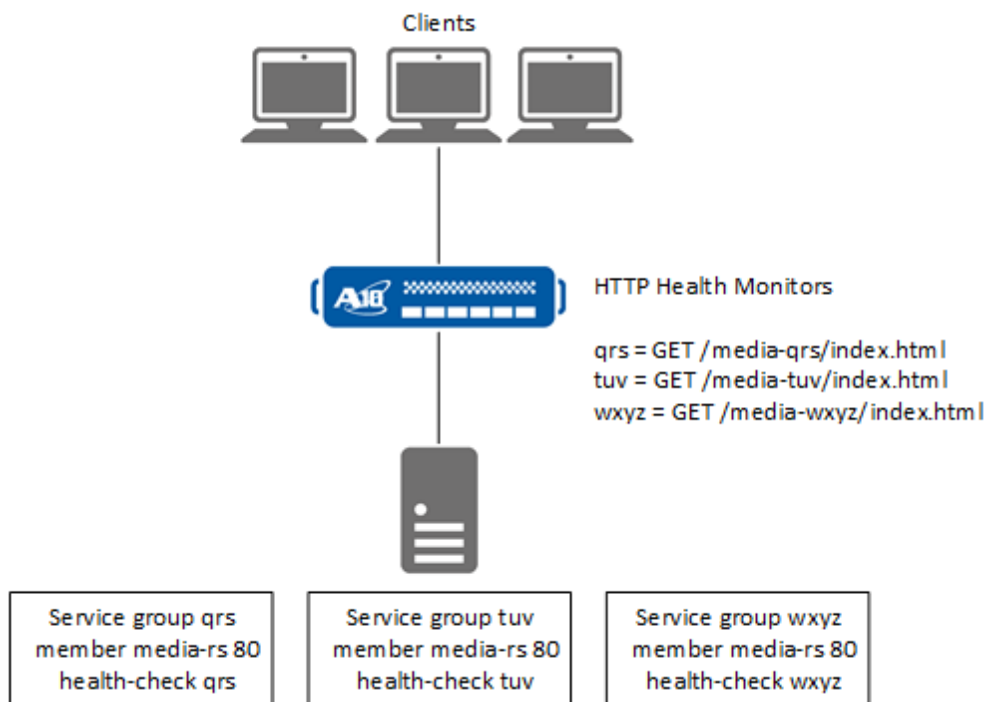
```

Service Group Health Checks

You can assign a health monitor to a service group. This is useful in cases where the same server provides content for multiple, independent sites. When using the feature, if a site is unavailable (for example, taken down for maintenance), the server fails the health check for that site, and clients are not sent to the site. However, other sites on the same server pass their health checks, and clients of those sites are sent to the server.

[Figure 44](#) shows an example.

Figure 44 : Service Group Health Checks



In this example, a single server provides content for the following sites:

- www.media-rtts.com
- www.media-tuv.com
- www.media-wxyz.com

All sites can be reached on HTTP port 80 on the server. The health check configured on the port in the real server configuration results in the same health status for all three sites. All of them either are up or are down.

In this case, if one of the sites is taken down for maintenance, the health status of that site will still be up, since the real port still responds to the health check configured on the port.

You can configure the ACOS device to separately test the health of each site, by assigning each site to a separate service group, and assigning a separate Layer 7 health monitor to each of the service groups. In this case, if a site is taken down for maintenance, that site fails its health check while the other sites still pass their health checks, on the same real port.

In this example, a separate HTTP health method is configured for each of the services. The health monitors test the health of a site by sending an HTTP request to a URL specific to the site. In this way, even though the server's HTTP port is up, a site will fail its health check if the URL requested by its health check is unavailable.

Priority of Health Checks

Service group health status applies only within the context of the service group. For example, a health check of the same port from another service group can result in a different health status, depending on the resource requested by the health check.

Health checks can be applied to the same resource (real server or port) at the following levels:

- In a service group that contains the real server and port as a member
- In a server or server port configuration template that is bound to the real server or port
- Directly on the individual real server or port

In cases where health checks are applied at multiple levels, they have the following priority:

- Health check on real server
- Health check on real server's port
- Health check on service group

If a health check at the real server level (1) fails, the corresponding real server, real server port, and service group members are marked Down. However, if a health check on the service group level (3) fails, only that service group member in that service group is marked Down.

To assign a health monitor to a service group, use either of the following methods.

Configuring Service Group Health Checks

GUI Configuration

Navigate to **ADC >> SLB >> Service Groups >> Update**, select the **monitor** from the drop-down list in the Health Monitor field.

CLI Configuration

The commands in this section implement the configuration shown in [Service Group Health Checks](#).

The following commands configure the health monitors for each site on the server:

```
ACOS(config)# health monitor qrs
ACOS(config-health:monitor)# method http url GET /media-qrs/index.html
ACOS(config-health:monitor)# exit
ACOS(config)# health monitor tuv
ACOS(config-health:monitor)# method http url GET /media-tuv/index.html
ACOS(config-health:monitor)# exit
ACOS(config)# health monitor wxyz
ACOS(config-health:monitor)# method http url GET /media-wxyz/index.html
ACOS(config-health:monitor)# exit
```

The following commands configure the real server:

```
ACOS(config)# slb server media-rs 10.10.10.88
```

```
ACOS(config-real server)# port 80 tcp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
```

The following commands configure the service groups:

```
ACOS(config)# slb service-group qrs tcp
ACOS(config-slb svc group)# health-check grs
ACOS(config-slb svc group)# member media-rs 80
ACOS(config-slb svc group-member:80)# exit
ACOS(config-slb svc group)# exit
ACOS(config)# slb service-group tuv tcp
ACOS(config-slb svc group)# health-check tuv
ACOS(config-slb svc group)# member media-rs 80
ACOS(config-slb svc group-member:80)# exit
ACOS(config-slb svc group)# exit
ACOS(config)# slb service-group wxyz tcp
ACOS(config-slb svc group)# health-check wxyz
ACOS(config-slb svc group)# member media-rs 80
ACOS(config-slb svc group-member:80)# exit
ACOS(config-slb svc group)# exit
```

The following commands configure the virtual servers:

```
ACOS(config)# slb virtual-server media-qrs 192.168.1.10
ACOS(config-slb vserver)# port 80 http
ACOS(config-slb vserver-vport)# service-group qrs
ACOS(config-slb vserver-vport)# exit
ACOS(config-slb vserver)# exit
ACOS(config)# slb virtual-server media-tuv 192.168.1.11
ACOS(config-slb vserver)# port 80 http
ACOS(config-slb vserver-vport)# service-group tuv
ACOS(config-slb vserver-vport)# exit
ACOS(config-slb vserver)# exit
ACOS(config)# slb virtual-server media-wxyz 192.168.1.12
ACOS(config-slb vserver)# port 80 http
ACOS(config-slb vserver-vport)# service-group wxyz
ACOS(config-slb vserver-vport)# exit
ACOS(config-slb vserver)# exit
```

To clear a port protocol session immediately if a member of a real service group level fails, enter the **del-session-on-server-down** command when configuring the port of

that member. For the above example, the configuration of the media-rs real server would be as follows:

```
ACOS(config)# slb server media-rs 10.10.10.88
ACOS(config-real server)# port 80 tcp
ACOS(config-real server-node port)# del-session-on-server-down
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
```

Disable Failed Health Check

You can configure the ACOS device to disable a server, port, or service group if the server, port, or service group fails a health check. This option is only available through the CLI.

This option applies to all servers, ports, or service groups that use the health monitor. When a server, port, or service group is disabled based on this command, the server, port, or service group's state is changed to **disable** in the running-config. If you save the configuration while the server, port, or service group is disabled, the state change is written to the startup-config.

ACOS also generates a log message to indicate that the server, port, or service group is disabled. The following is an example of a log message generated from the health monitor process:

```
Sep 20 2017 10:07:52 Warning [HMON]:Server s250 (192.168.2.250) is
disabled due to health check failure.
```

The server, port, or service group remains disabled until you explicitly enable it. Because a port is "enabled" by default, you may need to explicitly disable the port with a CLI command before re-enabling the port after a health check failure. The following CLI command perform that function:

```
ACOS(config-real server)# disable
ACOS(config-real server)# enable
```

Configure a Failed Health Check to Disable a Component (CLI Example)

Use the following commands:

```
ACOS(config)# health monitor hml
```

```
ACOS(config-helath:monitor)# disable-after-down
```

In-Band Health Monitoring

In-band health checks are an optional supplement to the standard Layer 4 health checks. In-band health checks assess service port health based on client-server traffic, and can very quickly send a client's traffic to another server and port if necessary. An in-band health check can also mark a port down.

In-band health monitoring is supported for the following service types:

- TCP
- HTTP
- HTTPS

Relationship To Standard Layer 4 Health Monitoring

The in-band health check works independently of and supplements the standard Layer 4 health check. For example, for TCP, the standard health check works as follows by default:

Every 30 seconds, the ACOS device sends a connection request (TCP SYN) to the specified TCP port on the server. The port passes the health check if the server replies to the ACOS device by sending a TCP SYN ACK.

This is the same Layer 4 health check available in previous releases and has the same defaults.

In-band health monitoring works as described below.

It is recommended that you continue to use standard Layer 4 health monitoring even if you enable in-band health monitoring. Without standard health monitoring, a server port marked down by an in-band health check remains down.

How In-Band Layer 4 Health Monitoring Works

In-band health monitoring for services on TCP watches client-server SYN handshake traffic, and increments the following counters if the server does not send a SYN ACK in reply to a SYN:

- **Retry counter** – Each client-server session has its own retry counter. the ACOS device increments a session's retry counter each time a SYN ACK is late. If the retry counter exceeds the configured maximum number of retries allowed, the ACOS device sends the next SYN for the session to a different server. ACOS also resets the retry counter to 0.

You can set the retry counter to 0-7 retries. The default is 2 retries.

- **Reassign counter** – Each real port has its own reassign counter. Each time the retry counter for any session is exceeded, the ACOS device increments the reassign counter for the server port. If the reassign counter exceeds the configured maximum number of reassignments allowed, the ACOS device marks the port DOWN.

In this case, the port remains DOWN until the next time the port successfully passes a standard health check. Once the port passes a standard health check, the ACOS device starts using the port again and resets the reassign counter to 0.

You can set the reassign counter to 0-255 reassignments. The default is 25 reassignments.

In-band health monitoring is disabled by default.

Logging and Traps

When the ACOS device marks a server port down, the device generates a log message and an SNMP trap, if logging or SNMP traps are enabled. The message and trap are the same as those generated when a server port fails a standard health check. However, you can discern whether the port was marked down due to a failed in-band health check or standard health check, based on the process name listed in the message.

- A10lb – The port was marked down by an in-band health check.
- A10hm – The port was marked down by a standard health check.

Here is an example of a log message generated from each process:

```
Sep 08 2014 17:15:04 Info      A10LB SLB server s-3-2-1 (10.3.2.1) port 80  
is down.  
Sep 08 2014 17:15:04 Info      A10HM SLB server s-3-2-1 (10.3.2.1) port 80  
is down.
```

In-band health monitoring does not mark ports up. Only standard health monitoring marks ports up. So messages and traps for server ports coming up are generated only by the A10hm process.

Configuring In-Band Health Monitoring

To configure in-band health monitoring:

1. Enable the feature in a server port template.
2. Bind the port template to real server ports, either directly or in a service group.

GUI Configuration

To configure in-band health monitoring in server port template:

1. Navigate to **ADC >> Template**.
2. Click the **Edit** link in the Actions column for an existing template, or click **Create** and select **Port** from the drop-down list to create a new port template.
3. Select the checkbox in the Inband Health Check field.
4. Enter other parameters as needed (for example, the template name, if you are creating a new template).
5. Click **OK** or **Update**.

To bind the template to a server port, see [Binding a Server Port Template to a Real Server Port](#).

CLI Configuration

To configure in-band health monitoring, use the `inband-health-check` command at the configuration level for the server port template:

The following example enables in-band health monitoring in a server port template called `rp-tmpl2` and binds this template to real ports 80 on server `rs1`, and real port 80 on server `rs2`:

```
ACOS(config)# slb template port rp-tmpl2
ACOS(config-rport)# inband-health-check
ACOS(config-rport)# exit
ACOS(config)# slb server rs1 10.1.1.99
ACOS(config-real server)# port 80 tcp
ACOS(config-real server-node port)# template port rp-tmpl2
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
ACOS(config)# slb server rs2 10.1.1.100
ACOS(config-real server)# port 80 tcp
ACOS(config-real server-node port)# template port rp-tmpl2
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
```

Consecutive Health Checks

You can configure the number of times the target device must consecutively reply to the same periodic health check in order to pass the health check.

By default, a server or port needs to successfully reply to a given health check only one time in order to pass the health check. The server or port is then considered to be up until the next periodic health check. You can set the required number of consecutive passes to 1-63.

You can configure this parameter on an individual health monitor basis. The setting applies to all health checks that are performed using the health monitor.

Consecutive Health Checks Within a Health Check Period Using the GUI

1. Navigate to **ADC >> Health Monitors**.
2. Click the Edit button next to the monitor name or click **Create** to add a new one.
3. If new, in the General Fields section, enter a name for the monitor.
4. If new, in the Method Type section, select the monitor type from the drop-down list, and enter or select settings for the monitor.
5. Select the check box in the Passive field.
6. In the Passive Interval field, enter the number of consecutive times the target must pass the same periodic health check.
7. Click **Create Monitor** or **Update Monitor**.
8. Apply the monitor to the server or server port.

Consecutive Health Checks Within a Health Check Period (CLI Example)

To configure consecutive health checks using the CLI, use the up-retry command. For example:

```
ACOS(config)# health monitor hml
ACOS(config-health:monitor)# up-retry 3
ACOS(config)# slb server ser1 172.168.1.112
ACOS(config-real server)# port 80 tcp
ACOS(config-real server-node port)# health-check hml
```

Maintenance Health Checks

You can use the response to an HTTP or HTTPS health check to set a server's health status to "Maintenance". When a server's health status is Maintenance, the server will accept new requests on existing cookie-persistent or source-IP persistent connections, but will not accept any other requests.

To place a server into maintenance mode, configure an HTTP or HTTPS health method that includes a maintenance code. If the server replies to a health check with the code, the ACOS device changes the server's health status to Maintenance.

To leave maintenance mode, the server must do one of the following:

- Successfully reply to a health check, but without including the maintenance code. In this case, the server's health status changes to Up.
- Fail a health check. In this case, the server's status changes to Down.

The Maintenance health status applies to server ports and service-group members. When a port's status changes to Maintenance, this change applies to all service-group members that use the port.

NOTE: This feature applies only to servers in cookie-persistence or source-IP persistence configurations, and can be used only for HTTP and HTTPS ports.

GUI Configuration

1. Navigate to **ADC >> Health Monitors**.
2. Click **Edit** to the modify the monitor or click **Create** to add a new one.
3. If new, in the General Fields section, enter a name for the monitor.
4. If new, in the Method Type section, select the monitor type (HTTP or HTTPS) from the drop-down list, and enter or select settings for the monitor.
5. In the HTTP or HTTPS fields configure the maintenance code.
6. Click **Create Monitor** or **Update Monitor**.
7. If this is a new Health Monitor, apply the monitor to the server or server port.

CLI Configuration

Use the maintenance-code *code-list* option with one of the following commands at the configuration level for a health method:

- **http** *options*
- **https** *options*

The following commands configure a health monitor that places a server into maintenance mode if the server replies to a health check with code 503:

```
ACOS(config)# health monitor hm2
ACOS(config-health:monitor)# method http maintenance-code 503
```

In this example, if the server replies with status 503, the server goes into maintenance mode and stays there until the server either fails a health check (Down) or replies with code 200 (Up). The following commands bind the health monitor to the real server port:

```
ACOS(config)# slb server ser2 172.168.1.114
ACOS(config-real server)# port 89 tcp
ACOS(config-real server-node port)# health-check hm2
```

On-Demand Health Checks

You can easily test the health of a server or individual service at any time, using the default Layer 3 health monitor (ICMP ping) or a configured health monitor.

Configuring On-Demand Health Checks (CLI Procedure)

To test the health of a server, use the **health-test** command at the EXEC, Privileged EXEC, or global configuration level of the CLI:

If an override IP address and protocol port are set in the health monitor configuration, the ACOS device uses the override address and port, even when an address and port is specified when the on-demand health check is sent.

Configuring On-Demand Health Checks (CLI Example)

The following command tests port 80 on server 192.168.1.66, using configured health monitor hm80:

```
ACOS# health-test 192.168.1.66 monitorname hm80
node status UP.
```

Enabling Strict Retries

Some types of health monitors do not use the retry parameter by default. For these health monitors, the ACOS device marks the server or port Down after the first failed health check attempt, even if the retries option for the health monitor is set to higher than 0.

For example, this is true for HTTP health monitors that expect a string in the server reply. If the server's HTTP port does not reply to the first health check attempt with the expected string, the ACOS device immediately marks the port Down.

To force the ACOS device to wait until all retries are unsuccessful before marking a server or port down, enable strict retries. You can enable strict retries on an individual health monitor basis.

GUI Configuration

1. Navigate to **ADC >> Health Monitors**.
2. Click **Edit** to the modify the monitor or click **Create** to add a new one.
3. In the General Fields section, select the checkbox for the **Strict Retry On Server Err Resp** field.
4. Click **Create Monitor** or **Update Monitor**.
5. If this is a new **Health Monitor**, apply the monitor to the server, server port, or service group.

CLI Configuration

The following example configures an HTTP health monitor that checks for the presence of "testpage.html", and enable strict retries for the monitor.

```
ACOS(config)# health monitor http-exhaust
ACOS(config-health:monitor)# method http url GET /testpage.html
ACOS(config-health:monitor)# strict-retry-on-server-err-resp
```

The following commands bind the health monitor to the real server port:

```
ACOS(config)# slb server ser3 172.168.1.115
ACOS(config-real server)# port 80 tcp
ACOS(config-real server-node port)# health-check http-exhaust
```

Globally Changing Health Monitor Parameters

You can globally adjust health monitor parameters, including the following:

- Interval – Number of seconds between health check attempts.
- Timeout – Number of seconds the ACOS device waits for a reply to a health check.
- Retries – maximum number of times the ACOS device will send the same health check to an unresponsive server or service before marking that server or service as down.
- Up-Retry – Number of consecutive times the device must pass the same periodic health check, in order to be marked Up.

Globally changing a health monitor parameter changes the default for that parameter. For example, if you globally change the interval from 5 seconds to 10 seconds, the default interval becomes 10 seconds.

If a parameter is explicitly set on a health monitor, globally changing the parameter does not affect the health monitor. For example, if the interval on health monitor hm1 is explicitly set to 20 seconds, the interval remains 20 seconds on hm1 regardless of the global setting.

Global health monitor parameter changes automatically apply to all new health monitors configured after the change. To apply a global health monitor parameter change to health monitors that were configured before the change, you must reboot the ACOS device.

When auto-adjust mode for health-check rate limiting is enabled and the global interval or timeout setting differs from the setting on an individual health monitor, actual interval and timeout values used depend on the number of health checks performed by the ACOS device. See [Health-Check Rate Limiting](#).

To globally change health monitor parameters, use the `health global` command.

Globally Changing Health Monitor Parameters (CLI Example)

The following commands globally changes the default number of retries to 5:

```
ACOS(config)# health global
ACOS(config-health:global)# retry 5
```

These commands globally change the interval and timeout to 10 seconds and default number of retries to 4:

```
ACOS(config)# health global
ACOS(config-health:global)# interval 10 timeout 10
```



```
ACOS(config-health:global) # retry 4  
ACOS(config-health:global) # exit
```

Globally Disabling Layer 3 Health Checks

Layer 3 health checks (ICMP pings) are enabled by default. For large deployments (1000 or more servers), A10 Networks, Inc. recommends disabling the default Layer 3 health check, and using only Layer 4-7 health checks.

To globally disable Layer 3 health checks, disable health monitoring in the server templates used to configure the servers. If you did not configure a customized server template, the default server template is used.

If a custom Layer 3 health monitor is enabled on an individual server, the health monitor is still used.

Globally Disabling Layer 3 Health Checks (CLI Example)

Use the **health-check-disable** command to disable Layer 3 health monitoring in the template:

```
ACOS(config) # slb template server default  
ACOS(config-rserver) # health-check-disable
```

Health-Check Rate Limiting

Health-check rate limiting helps ensure health checks in large SLB deployments can be processed successfully. In previous releases, without health-check rate limiting, it is possible for a server or port to fail its health check because the ACOS device is unable to complete processing of the health check before it times out.

Health-check rate limiting is enabled by default. Generally, you do not need to configure it. However, you still may want to see the following section: [Health-Check Guidelines](#).

Health-check rate limiting uses a threshold to limit the number of health-check packets the ACOS device will send within a given 500-millisecond (ms) period. If additional health-check packets need to be sent, the ACOS device waits until the next

500-ms period. Within any 500-ms period, the ACOS device never sends more health-check packets than are allowed by the threshold.

Health-check rate limiting has an auto-adjust mode, which is enabled by default. When necessary, the auto-adjust mode dynamically increases the default interval and timeout for health checks. By increasing these timers, health-check rate limiting provides more time for health-check processing.

Health-check rate limiting is always enabled, and its auto-adjust mode is enabled by default.

Health-check Threshold

When auto-adjust mode is enabled, the health-check threshold is set to the following:

- 1600 health-check packets per 500-ms period.

When auto-adjust mode is enabled, you can not manually change the threshold. To change the threshold, you first must disable auto-adjust mode. Then you can set the threshold to a value within one of the following ranges:

- 1 - 5000 health-check packets per 500-ms period.

When you disable auto-adjust mode, the default threshold is changed to one of the following:

- 1000 health-check packets per 500-ms period.

NOTE:	It is recommended not to set the threshold to a very high value. Doing so may result in health-check timeouts due to resource constraints.

Health-Check Interval and Timeout

When the auto-adjust mode for health-check rate limiting is enabled, and the global interval or timeout setting differs from the setting on an individual health monitor, the actual interval and timeout values that are used depend on the number of health checks performed by the ACOS device:

- More than 1000 health checks – Setting with the longer value is used. For example, if the global interval is 10 seconds but the interval configured on an individual health monitor is 5 seconds, 10 seconds is used.
- Fewer than 1000 health checks – Setting on the individual health monitor is used.

Configuring Health-Check Rate Limiting

Displaying Health-Check Rate Limiting (CLI Example)

To display the current settings for health-check rate limiting, use the **show health stat** command. Here is an example:

```
ACOS# show health stat
Health monitor statistics
Total run time:                : 21 hours 31 minutes 31 seconds
Number of burst:                : 0
...
Configured health-check rate (/500ms) : Auto configured
Current health-check rate (/500ms):    : 5
...
```

The Configured health-check rate field and Current health-check rate show the health-check rate limiting settings.

- If auto-adjust is enabled:
 - Configured health-check rate – Shows “Auto configured”
 - Current health-check rate – Shows the total number of health monitors divided by the global health-check timeout:
`total-monitors / global-timeout`
- If auto-adjust is disabled:
 - Configured health-check rate – Shows the manually configured threshold
 - Current health-check rate – Shows the manually configured threshold

Changing the Threshold (CLI Example)

To change the health-check rate limiting threshold:

```
ACOS(config)# health global  
ACOS(config-health:global)# disable-auto-adjust  
ACOS(config-health:global)# check-rate 2000
```

Multi-CPU Support for Health Monitoring

ACOS includes a scalability feature that enables use of multiple CPUs for health-monitor processing.

Ensure that the health-monitor process number does not exceed the current number of control CPUs. The maximum value for the health check-rate is 50000000.

GUI Configuration

To configure multi-CPU support from the GUI:

1. Navigate to **ADC >> Health Monitors >> Global**.
2. Select the number of CPUs from the drop-down list in the Multi Process field.
3. Click **Update**.

CLI Configuration

Use the following commands to specify 2 processes to start health monitoring.

```
ACOS(config)# health global  
ACOS(config-health:global)# multi-process 2
```

You can monitor up to 20 processes, with the default value being 1.

It is recommend that you use this feature with multiple control CPUs. ACOS should be configured to run multiple control CPUs before issuing the health multi-process command. For example:

```
ACOS(config)# multi-ctrl-cpu 2  
This will modify your boot profile for multiple control CPUs.  
It will take effect after the next reboot.  
Please confirm: You want to configure multiple control CPUs (N/Y)?: y
```

Database Health Monitors

You can configure database health monitors using the CLI or GUI, without the need to configure and import scripts. Previous releases support database health monitors only with imported scripts. Simplified database health monitor configuration applies to the following database types:

- Oracle
- MSSQL
- MySQL
- PostgreSQL

Database Health Monitor Parameters

To configure a database health monitor, specify the following parameters.

Required Health Monitor Parameters

- Database type – Oracle, MSSQL, MySQL, or PostgreSQL
- Database name – Name of the database to query
- Username and password – Account information required for access to the database

Optional Health Monitor Parameters

- Send string – SQL query to send to the database.
- Receive string – Query result expected from the database in order to pass the health check.
- Receive row and column – For replies that consist of multiple results, the results are in a table. You can specify the row and column location within the results table to use as the receive string. If you do not specify the row and column, row 1 and column 1 are queried by default.

To use the receive string option, you also must use the send string option. If you do not use the send option, the ACOS device does not send a query.

Configuring Database Health Monitors

GUI Configuration

To create a new database health monitor:

1. Navigate to **ADC >> Health Monitors**.
2. Select **Create**.
3. In the Name field, specify a name for your health monitor.
4. In the Method type field, select **Database** from the drop-down list.
5. Configure the desired values in the General Fields pane.
6. In the Database pane, select the database type from the Database Type field, then complete the other parameters.
7. Click **Create Monitor**.

CLI Configuration

The example in this section creates an Oracle database health monitor:

```
ACOS(config)# health monitor dbhml  
ACOS(config-health:monitor)# method database oracle db-name orcl1 username  
admin password welcome1
```

For additional information, see [Database Health Monitor Parameters](#).

Kerberos Health Monitors

ACOS provides health monitoring for Kerberos AAA servers. A Kerberos health monitor can contain separate methods for checking accessibility to each of the following:

- Key Distribution Center (KDC) Ticket Granting Ticket (TGT) service – Tests ability to access the Kerberos server as a new client requesting a TGT.
- Kerberos administrative system – Tests ability to access the Kerberos server as a user account administrator.

- Kerberos password change system – Tests ability to access the Kerberos server as a user attempting to change their password.

These services can be on the same server (hostname or IP address) or different servers.

You can use these health checks in AAA SLB deployments and in Authentication Proxy deployments.

Health checks for access to the Kerberos administrative system are not supported with Windows Server Domain Controllers (DCs).

GUI Configuration

1. Navigate to **ADC >> Health Monitors**.
2. Click **Create**.
3. Specify **kerb-hm1** in the Name field.
4. Select “Kerberos KDC” from the drop-down list in the Method type field.
5. Complete the necessary fields in the Kerberos KDC field
6. Click **Create Monitor**.

CLI Configuration

To configure a Kerberos health monitor, use the commands described in this section.

The following commands create a Kerberos health check method to check accessibility of the KDC for obtaining a TGT (“acos-1” is the name of the ACOS device):

```
ACOS(config)# health monitor kerb-hm1
ACOS(config-health:monitor)# method kerberos-kdc kinit acos-1
examplepassword examplekdchost
```

The following command configures a Kerberos health check method to check accessibility of the Kerberos server for user account administration (“acos-1” is the name of the ACOS device):

```
ACOS(config-health:monitor)# method kerberos-kdc kadmin exemplerealm acos-1
examplepassword examplekdchost exampleadminhost
```

The following example configures a Kerberos health method to check accessibility of the Kerberos server for user password changes (“acos-1” is the name of the ACOS device):

```
ACOS(config-health:monitor)# method kerberos-kdc kpasswd acos-1  
examplepassword examplekdchost examplepwdhost
```

Compound Health Monitors

Compound health monitors enable you to combine the status of multiple health checks into a single result. ACOS uses the combined results of individual health checks to determine the health of the real server, port, or service group to which the compound health monitor is applied.

- The compound server health status may report as Down due to incorrect timeout or interval parameters. (See [Compound Health Monitor Considerations](#))
- Compound health monitoring increases the workload of the health monitoring subsystem. For example, using a compound monitor with many sub monitors against a service group with many members can affect system performance. This increased workload could significantly delay the response time for compound health checks.

Compound Health Monitor syntax

A compound health monitor consists of a set of health monitors joined in a Boolean expression (AND / OR / NOT). The CLI Boolean expression syntax is Reverse Polish Notation (also called Postfix Notation), a notation method that places an operator (AND, OR, NOT) after all of its operands (in this case, health monitors).

After listing the health monitors, add the Boolean operator(s). The following operators are supported:

- AND – Both the ANDed health checks must be successful for the health status to be Up. If either of the health checks is unsuccessful, the health status is Down.
- OR – Either of the ORed health checks must be successful for the result to be Up. Even if one of the health checks is unsuccessful, the health status is still Up if the

other health check is successful. If both of the health checks are unsuccessful, the health status is Down.

- NOT – The health status is the opposite of the health check result. For example, if a health check is unsuccessful, the resulting health status is Up. Likewise, if the health check is successful, the resulting health status is Down. You can use NOT with a single health method, or with multiple health methods for more complex expressions. (See below.)

For example, to construct a health monitor that ANDs two health monitors, use the following syntax:

```
method compound sub hm1 sub hm2 and
```

This is logically equivalent to the following expression: `hm1 & hm2`

- In the CLI, you must enter `method compound` at the beginning, and `sub` in front of each health-monitor name. In the GUI, do not enter `method compound`. The GUI automatically enters `sub` in front of each health monitor name when you select it.
- The equivalent expressions are shown for clarity but are not valid syntax on the ACOS device.

Similarly, to construct a health monitor that ORs two health monitors, use the following syntax:

```
method compound sub hm1 sub hm2 or
```

This is logically equivalent to the following expression: `hm1 | hm2`

To construct a health monitor that results in an Up health status if the health check is *unsuccessful*, use the following syntax:

```
method compound sub hm1 NOT
```

This is logically equivalent to the following expression: `! hm1`

To construct more complex expressions, you can enter multiple sets of health monitors and operators. Here is a quite complex expression:

```
(! (hm1 | (hm2 & (hm3 | (! hm4)))) | hm5
```

To configure this expression, use the following syntax:

```
method compound sub hm1 sub hm2 sub hm3 sub hm4 not or and or not sub hm5
or
```

Compound Health Monitor Considerations

- A maximum of 8 sub monitors are supported in a compound monitor. To use more sub monitors, you can nest compound monitors. (See below.)
- The total number of sub monitors plus the number of Boolean operators supported in a compound monitor is 16.
- You can nest compound monitors. To nest compound monitors, configure a compound monitor and use that monitor as a sub monitor in another compound monitor. The maximum nesting depth is 8.

Nesting loops are not allowed.

- The `timeout` and `interval` parameters of a compound monitor must be set to values that allow each of the sub monitors to complete their health checks. If any of the sub monitors is unable to complete its health check, the compound monitor's result is always Down. The formula for calculating the compound health monitor is below:

```
timeout (compound monitor) > interval x retry + timeout (sub monitor)
```

For example, if monitor1 gives a result after 2 seconds, but a compound monitor that uses monitor1 times out in 1 second, the resulting health status will always be Down, regardless of the Boolean expression. (See [Health Monitors – Timeout](#).)

If you encounter this problem, A10 Networks recommends you increase the `timeout` parameter for the compound monitor to ensure the health check can cycle through all sub monitors. (See [Configuring and Applying a Health Method](#).) You alternatively can decrease the number of retry attempts for sub monitor health checks to 1. (See [Consecutive Health Checks](#).)

Configuring Compound Health Monitors

GUI Configuration

1. Navigate to **ADC >> Health Monitor**.
2. Configure the sub monitors first.
3. Click **Create** on the health monitor list to begin configuration of a new monitor.

4. In the Method Type section, select **Compound** from the drop-down list. The Compound configuration controls appear.
5. Construct the Boolean expression:

Use Reverse Polish Notation. (See [Compound Health Monitor syntax](#).) Otherwise, the GUI displays an error message when you click OK to complete the health monitor configuration.
6. Click **Create Monitor** to complete the configuration of the compound monitor.

CLI Configuration

To configure a compound health monitor, use the `method compound` command to add a Boolean expression at the configuration level for the monitor:

Use Reverse Polish Notation. (See [Compound Health Monitor syntax](#).)

The following commands configure a compound health monitor in which both health checks must be successful in order for the resulting health status to be Up:

```
ACOS(config)# health monitor hm-compoundAND
ACOS(config-health:monitor)# method compound sub hm1 sub hm2 and
```

The following commands apply the health monitor to a service group:

```
ACOS(config)# slb service-group sg1 tcp
ACOS(config-slb svc group)# health-check hm-compoundAND
```

The following commands configure a compound health monitor in which the resulting health status is Up if any one or more of the health checks is successful:

```
ACOS(config)# health monitor hm-compoundOR
ACOS(config-health:monitor)# method compound sub hm1 sub hm2 sub hm3 or or
```

The following commands apply the health monitor to a service group:

```
ACOS(config)# slb service-group sg2 tcp
ACOS(config-slb svc group)# health-check hm-compoundOR
```

Health Monitors – NOT Operator

The following commands configure a compound health monitor that results in an Up status *only* if the server *fails* the ICMP health check:

```
ACOS(config)# health monitor icmp
ACOS(config-health:monitor)# retry 1
ACOS(config-health:monitor)# exit
ACOS(config)# health monitor not-ping
ACOS(config-health:monitor)# interval 11 timeout 11
ACOS(config-health:monitor)# method compound sub icmp not
```

Health Monitors – Timeout

The following commands configure a compound health monitor that results in **Down** status if either monitor1 or monitor2 *fails* its health check, because the compound monitor uses the **and** operator:

```
ACOS(config)# health monitor monitor1
ACOS(config-health:monitor)# interval 3 timeout 3
ACOS(config-health:monitor)# retry 2
ACOS(config-health:monitor)# method tcp port 80
ACOS(config-health:monitor)# exit
ACOS(config)# health monitor monitor2
ACOS(config-health:monitor)# interval 5 timeout 5
ACOS(config-health:monitor)# retry 1
ACOS(config-health:monitor)# method tcp port 8080
ACOS(config-health:monitor)# exit
ACOS(config)# health monitor compound-monitor
ACOS(config-health:monitor)# interval 6 timeout 6
ACOS(config-health:monitor)# retry 3
ACOS(config-health-monitor)# method compound sub monitor1 sub monitor2 and
```

The amount of time allowed to perform health checks for monitor1 and monitor2 is calculated by the retry and timeout values.

For this example, a compound health check requires a minimum of 10 seconds. If the timeout value for the compound health check is set to 6, then the result is always Down. In order to ensure a complete compound health check, set the compound health check timeout to 10 seconds or greater.

Configurable Response Codes: RADIUS Health Monitors

By default, ACOS requires a RADIUS server to reply to RADIUS health checks with an Access-Accept message (response code 2, by default).

You can specify the response code(s) that ACOS can accept as valid responses. So, for example, you can configure a health monitor to accept an Access-Reject response (code 3) as a valid response from a healthy RADIUS server.

To specify the response code(s) a RADIUS server is allowed to send back in response to a RADIUS health check, use the **expect response-code** option with the **method** command.

Configuring Response Codes: RADIUS Health Monitors (CLI Example)

The following commands configure a RADIUS health monitor that accepts response code 2 or 3 as passing (healthy) responses from a server:

```
ACOS(config)# health monitor rad1
ACOS(config-health:monitor)# method radius username admin1 password
examplepassword secret a10rad port 1812 expect response-code 2,3
```

Displaying Health Status

ACOS begins sending health checks to a real server's IP address and service ports as soon as you finish configuring the server. You can display health status for virtual servers and service ports, and for the real servers and service ports used by the virtual server.

This section describes the methods to display health status.

The following topics are covered:

Viewing Health of Virtual Servers and Service Ports	386
Viewing Health of Real Servers and Service Ports	387
Viewing Health Monitoring Statistics	388

Viewing Health of Virtual Servers and Service Ports

GUI Configuration

1. Navigate to **ADC >> SLB**.

The virtual servers are listed at the top of the page. An icon appears next to each virtual server to indicate the health status.

2. To display specific health information for a virtual server's service ports, click the virtual server name.

CLI Configuration

Use the **show slb virtual-server** command to view the health of virtual servers and service ports. The health status is shown in the State field; for additional information about this command and the fields in the output, see the *CLI Reference*.

```
ACOS# show slb virtual-server "vs 1"
Virtual server: vs 1(A)                               State: Down                               IP:
1.1.1.1.201
Port                                                    Curr-conn  Total-conn  Rev-Pkt    Fwd-Pkt
Peak-conn
-----
-----
Virtual Port:80 / service:svcgrp 1 / state:Down
  port 80  tcp
1          server 1:80/Down    0           0           0           0           0

Virtual Port:1 / service: / state:Unkn
  port 1   tcp
          Persist Source IP:tmpl persist srcip 1

Virtual Port:2 / service: / state:Unkn
  port 2   tcp
          Persist SSL session ID:tmpl persist sslid 1

Virtual Port:3 / service: / state:Unkn
  port 3   tcp
```

```
Template tcp tmp1 tcp 1
```

Viewing Health of Real Servers and Service Ports

GUI Configuration

1. Navigate to **ADC >> SLB >> Servers** tab.
2. An icon appears next to each real server to indicate the health status.
3. To display more specific health status information for a real server's service ports, click the server name.

For information about the real server health state icons, see the online help or *GUI Reference*.

CLI Configuration

Use the `show slb server` command to view the health of real servers and service ports. The health status is shown in the State column; for additional information about this command and the fields in the output, see the *Command Line Interface Reference*.

```
ACOS# show slb server
Total Number of Services configured: 5
          Current = Current Connections, Total = Total
Connections
          Fwd-pkt = Forward packets, Rev-pkt = Reverse packets
Service          Current    Total    Fwd-pkt    Rev-pkt
Peak-conn    State
-----
s1:80/tcp          0         0         0         0
          Down
s1:53/udp          0         0         0         0
          Down
s1:85/udp          0         0         0         0
          Down
s1: Total          0         0         0         0
          Down
```

...

Viewing Health Monitoring Statistics

Use the **show health stat** command to view health monitoring statistics.

```
ACOS# show health stat
Health monitor statistics
Total run time:                : 2 hours 1345 seconds
Number of burst:               : 0
max scan jiffie:               : 326
min scan jiffie:               : 1
average scan jiffie:           : 1
Opened socket:                 : 1140
Open socket failed:            : 0
Close socket:                  : 1136
Send packet:                   : 0
Send packet failed:            : 259379
Receive packet:                : 0
Receive packet failed          : 0
Retry times:                   : 4270
Timeout:                       : 0
Unexpected error:              : 0
Conn Immediate Success:        : 0
Socket closed before 17:       : 0
Socket closed without fd notify: : 0
Get retry send:                : 0
Get retry recv:                : 0
Configured health-check rate (/500ms) : Auto configured
Current health-check rate (/500ms):   : 1600
External health-check max rate (/200ms) : 2
Total number:                  : 8009
Status UP:                     : 8009
Status DOWN:                   : 0
Status UNKN:                   : 0
Status OTHER:                  : 0

IP address   Port   Health monitor   Status Cause(Up/Down)  Retry PIN
-----
10.0.0.11    80    http             UP      11 /0 @0             0      0 /0 0
```



```
10.0.0.12      80      http      UP      10 /0 @0      0      0 /0 0
```

Use the **show log | inc Sub-compound** command to view the health status of sub-compounds. This is applicable only for Compound health monitor logs.

```
Oct 28 2022 15:55:26 Info [HMON]:Sub-compound udp53 on the server
([fc00::101]:53/UDP) is up (UDP No Response)
Oct 28 2022 15:55:22 Info [HMON]:Sub-compound dns53 on the server
([fc00::101]:53/DNS) is down (SNMP Receive OK)
Oct 28 2022 15:55:22 Info [HMON]:Sub-compound tcp25 on the server
([fc00::101]:25/TCP) is up (TCP Verify Connection OK)
```

For additional information, see the *Command Line Interface Reference*.

Using External Health Methods

In addition to internal (predefined) health check methods, the system supports external health methods, which allow you to perform custom health checks using scripts.

Supported Script Types

The following types of scripts are supported:

- Perl
- Shell
- Tcl
- Python

Supported Capabilities

The following capabilities are supported and can be used within scripts:

- Perl - `IO::Socket` module for network communication
- Tcl - UDP extension for network-based checks
- Utility commands (such as `ping`, `ping6`, `wget`, `dig`, and `curl`)

Limitation

- External health methods are not supported in Direct Server Return (DSR) deployments.

Runtime Environment

External health monitor scripts execute within a restricted sandbox environment designed to protect system integrity while providing controlled access to approved runtime environments and utilities.

The runtime environment provides:

- Controlled execution context
- Restricted system access
- Approved runtime environments
- Approved command utilities
- Isolation from unrestricted system operations

By default, scripts execute with restricted privileges.

The following sections describe the supported runtime environments, available commands, runtime restrictions, and optional root-shell execution capabilities.

The following topics are covered:

Supported Runtime Environments and Commands	390
Using External Health Monitors with Root-Shell Access	393

Supported Runtime Environments and Commands

External health monitors support a controlled runtime environment that allows execution of approved shells, interpreters, and command-line utilities. Only the commands listed in this section are supported for use within external health monitor scripts.

Shells and Interpreters

The following shells and interpreters are supported for external health monitor scripts:

Supported Runtime	Category
<code>bash</code>	Shell
<code>/bin/bash</code>	Shell
<code>sh</code>	Shell
<code>tclsh</code>	Interpreter
<code>perl</code>	Interpreter
<code>python3</code>	Interpreter
<code>expect</code>	Automation

Network and Monitoring Commands

The following controlled set of shell utilities and command-line tools are supported for external health monitor scripts:

Command	Purpose
<code>arping</code>	Verify Layer-2 reachability
<code>curl</code>	Transfer data using supported protocols NOTE: For HTTPS connections, use the <code>-k</code> option to bypass SSL certificate verification when required. Use this option with caution.
<code>dig</code>	Perform DNS lookups Example: <code>dig @8.8.8.8 google.com</code>
<code>ip</code>	Display or manage network information
<code>nslookup</code>	Query DNS servers
<code>ping</code>	Verify network reachability

Command	Purpose
<code>snmpget</code>	Retrieve SNMP data
<code>snmpwalk</code>	Retrieve SNMP subtree information
<code>tftp</code>	Transfer files using TFTP
<code>traceroute</code>	Trace network path to a host
<code>wget</code>	Retrieve content using HTTP or HTTPS

System and Text Processing Commands

The following system and text processing commands are supported for external health monitor scripts:

Command	Command	Command	Command
<code>awk</code>	<code>echo</code>	<code>logger</code>	<code>sh</code>
<code>axlog</code>	<code>egrep</code>	<code>ls</code>	<code>sleep</code>
<code>base64</code>	<code>env</code>	<code>lspci</code>	<code>sort</code>
<code>basename</code>	<code>expect</code>	<code>md5sum</code>	<code>stat</code>
<code>bash</code>	<code>expr</code>	<code>mkdir</code>	<code>tar</code>
<code>cat</code>	<code>fgrep</code>	<code>mv</code>	<code>tclsh</code>
<code>chmod</code>	<code>file</code>	<code>paste</code>	<code>touch</code>
<code>cp</code>	<code>find</code>	<code>perl</code>	<code>tr</code>
<code>cut</code>	<code>free</code>	<code>ps</code>	<code>uniq</code>
<code>date</code>	<code>grep</code>	<code>python</code>	<code>uuidgen</code>
<code>df</code>	<code>gzip</code>	<code>python3</code>	<code>vi</code>
<code>diff</code>	<code>head</code>	<code>rm</code>	<code>vim</code>
<code>dir</code>	<code>ln</code>	<code>sed</code>	<code>wc</code>
<code>dirname</code>	<code>localtime</code>	<code>seq</code>	<code>xargs</code>

TCL Runtime Restrictions

To maintain system security and platform integrity, selected Tcl commands are restricted within the external health monitor runtime environment. Tcl commands that are not listed in this section remain available.

The following Tcl commands are blocked or restricted on the ACOS device in certain controlled environments:

Command	Command
<code>cd</code>	<code>namespace</code>
<code>clipboard</code>	<code>open</code>
<code>exec</code>	<code>package</code>
<code>exit</code>	<code>pwd</code>
<code>file</code>	<code>seek</code>
<code>flush</code>	<code>selection</code>
<code>glob</code>	<code>send</code>
<code>interp</code>	<code>source</code>
<code>load</code>	<code>toplevel</code>
<code>memory</code>	<code>update</code>

Using External Health Monitors with Root-Shell Access

ACOS supports execution of external health-monitor scripts within a controlled environment. By default, external scripts run in a restricted sandbox mode with limited system access. However, scripts can optionally be executed with elevated (root) privileges when the `root-shell` option is enabled.

Root-shell execution is available only when the **PRIVILEGE_SHELL_ROOT** license is installed. When this license is active, external scripts are executed with root privileges inside the sandbox environment. For the Shell Root license, contact [A10 Support](#).

NOTE:

- Currently, root-shell execution is supported for bash-based external scripts only.
- If the **PRIVILEGE_SHELL_ROOT** license is removed without a system reboot, previously configured health monitor scripts with `root-shell` enabled continue to run with elevated privileges until the next reboot.

CLI Configuration

To enable the `root-shell` option for an external health monitor:

```
ACOS(config-health:monitor)#method external program <program_name> root-shell
```

To verify that the root-shell option is applied:

```
ACOS(config-health:monitor)#show context
!Context configuration: 92 bytes
!
health monitor ext1_r
  interval 15 timeout 12
  method external program e1 root-shell
!
ACOS(config-health:monitor)#
```

Supported Root-Shell Commands

When `root-shell` is enabled, the following commands are available with root privileges inside the sandbox environment:

Table 9 : Supported Shell Commands

Command	Purpose
<code>arping</code>	Send ARP requests to a neighbor host to discover/verify MAC addresses
<code>awk</code>	Scan and process patterns in text
<code>axlog</code>	View or manage system logs (platform-specific)
<code>base64</code>	Encode or decode data in Base64 format

Table 9 : Supported Shell Commands

Command	Purpose
basename	Extract the file name from a full file path or directory
bash	Execute commands in the Bash shell
cat	Concatenate and display file content
chmod	Change file permissions
cp	Copy files and directories
curl	Transfer data to and from a server using different protocols. While using the <code>curl</code> command, you need to specify the <code>-k</code> option to bypass SSL certificate verification and allow the server connection.
cut	Remove sections from each line of files (extract columns)
date	Display or set the system date and time
df	Display disk space usage
diff	Compare files line by line
dig	Perform DNS lookups to query DNS name servers for information
dir	List directory contents
dirname	Extract directory path from file path
echo	Display a line of text or variable values passed as an arguments
egrep	Search using extended regex patterns
env	Display or set environment variables
expect	Automate interactive applications
expr	Evaluate expressions and performs arithmetic or string operations
fgrep	Search for fixed string patterns
file	Determine file type
find	Search for files in directories

Table 9 : Supported Shell Commands

Command	Purpose
free	Display memory usage
grep	Search for patterns within files or input streams
gzip	Compress files
head	Output the first part (usually 10 lines) of files
ip	Show or manipulate routing, network devices, interfaces, and tunnels
ln	Create links between files
logger	Add messages to system logs
ls	List all directories
lspci	List all PCI devices connected to the system
md5sum	Compute or check MD5 hash
mkdir	Create new directories
mv	Move or rename files and directories.
nslookup	Query DNS servers
paste	Merge lines from multiple files side by side
perl	Execute scripts written in the Perl programming language
ping	Send ICMP echo requests to network hosts to check connectivity
ps	Display running processes
python	Execute scripts written in the Python programming language
rm	Remove files or directories from the system
sed	Stream editor for filtering and transforming text
seq	Generate sequences of numbers
sh	Execute shell scripts
sleep	Pause or delay execution for a specified amount of time
snmpget	Communicate with a network entity using SNMP GET

Table 9 : Supported Shell Commands

Command	Purpose
	requests
<code>snmpwalk</code>	Retrieve SNMP data from devices
<code>sort</code>	Sort file contents
<code>stat</code>	Display detailed file or file system status
<code>tar</code>	Archive files
<code>tclsh</code>	Execute Tcl scripts using the Tcl shell command-line interface
<code>tftp</code>	Transfer files using TFTP protocol
<code>touch</code>	Create empty files or update timestamps
<code>tr</code>	Translate or delete characters
<code>traceroute</code>	Trace network path to a host
<code>uniq</code>	Filter duplicate lines
<code>uuidgen</code>	Generate UUIDs
<code>wc</code>	Count lines, words, characters
<code>wget</code>	Download files from web directly to an active directory
<code>xargs</code>	Build argument lists and execute commands

Security and Operational Controls

The following topics are covered:

Securing External Health Monitors	397
Restricting Extended Health Monitor Activity	398
Monitoring Extended Health Monitor Activity	399
Reviewing Extended Health Monitor Configuration	399

Securing External Health Monitors

Inherent in the External Health Monitor feature is the addition of programmatic code, in the form of scripts, by the customer to the closed, system-level, run-time

environment within ACOS. Running at elevated processing privilege, this code will have intimate access throughout the ACOS system and broad connectivity to the deployed networking environment.

Well behaved scripts contain trusted code solely for performing their functions to monitor the health of real servers and applications in the customers network. **It is the customer's responsibility to ensure the integrity and content of these scripts** – that the code is trusted and the scripts are free of code that can compromise the security of the ACOS system or systems in connected networks.

Malicious code in External Health Monitor scripts, whether intentional by disaffected ACOS admins or unintentional by compromised ACOS admins or their systems, can expose the ACOS system and connected environment to a range of exploits, such as, but not limited to, the following:

- Denial of Service (DoS), Distributed Denial of Service (DDoS)
- Remote Code Execution (RCE), Buffer Overflow, Brute Force
- Side-channel, exfiltration of Sensitive Information
- Man-in-the-Middle
- Spyware, Ransomware
- Trojan Horses, Network Worms

To secure External Health Monitors and keep them secure, ACOS administration can restrict access to these services, monitor access activity, and review the content of these scripts.

Restricting Extended Health Monitor Activity

All ACOS admins, except for the ACOS root admin, are not permitted to import, create, edit/modify, or delete External Health Monitor scripts as a default. Selected ACOS admins sufficiently trusted to access these scripts can be allowed to perform these operations by provisioning the administrative configuration with the health monitor (`hm`) privilege.

Only ACOS, system-level admins with Read-Write (R/W) privilege and specifically assigned this new privilege will be permitted to perform these operations for External Health Monitor scripts. As these monitoring scripts have access throughout

the ACOS system, this privilege is not available to partition constrained ACOS admins.

Best Practice: To minimize the threat surface of this feature, provision this privilege for the fewest users necessary.

For more information on configuring administrative users, refer to the *Management Access and Security Guide* or [Configuring External Health Monitors](#).

Monitoring Extended Health Monitor Activity

External health monitor activity can be monitored via the ACOS Audit Log. This log can be configured to log events to an external server through the `logging auditlog host` CLI command.

Script access operations can be detected on the logging server by filtering for the following strings and reporting match events for administrative review.

- `import health-external`
- `health external create`
- `health external edit`
- `health external delete`

Reviewing Extended Health Monitor Configuration

External health monitor scripts can be reviewed and audited periodically or upon detecting audit log activity to ensure their integrity and intended use in the ACOS system.

Instantiated scripts can be listed by the `show health external` ACOS CLI command and inspected individually using the `show health external file.ext` ACOS CLI command, where `file.ext` is one of the indicated script files listed.

Configuring External Health Monitors

HM privilege enables an administrator to access and manage External Health Monitor files. An account with write privilege that does not include HM access cannot import, edit, create, or delete External Health Monitor files.

To configure an external health monitor, you must first configure an admin account.

GUI Configuration

1. Navigate to the **System >> Admin >> Users >> Update**.
2. Enter the account name to be modified in the `username` field.
3. Verify that `Privilege type` is set to `Global`.
4. For `Privilege`, select `Read/Write/HM`
5. Save changes by clicking the `update` button.

To use an external health method:

1. Configure a health monitor script.
2. Import the script onto the ACOS device.
3. Configure a health monitor that uses external as the method.
4. In the server configuration, set the health check to use the method.

NOTE: The filename of the imported script should be less than 32 characters in length, including the extension.

To create an external health-check script onto the ACOS device:

1. Select **ADC > Health Monitors**.
2. On the menu bar, select **External Programs**.
3. Click **Create**.
4. Enter a name and description for the script.
5. Copy-and-paste the script into the **Definition** field.
6. Click **Create**.

To create an external health monitor:

1. Select **ADC > Health Monitors**.
2. Click **Create**.
3. In the **Name** field, enter a name for the monitor.
4. In the **Method Type** section, select **External** from the drop-down menu.
5. Click the **Program** drop-down menu and select the desired script.

6. In the **Port** field, enter the protocol port number.
7. Enter any arguments in the **Arguments** field.
8. Click **Create Monitor**.

To configure a server to use the external health method:

1. Select **ADC > SLB**.
2. On the menu bar, select the **Servers** tab.
3. Click the server name or click **Create** to create a new one.
4. If configuring a new server, enter the name and IP address, and other general parameters as applicable.
5. In the **Port** section:
 - a. If adding a new port, enter the port number in the Port field.
 - b. Select the external health monitor from the Health Monitor field.
6. Click **Create** or **Update**.

CLI Configuration

1. Specify the admin username using the `admin` command: A password is required when configuring a new admin account

```
ACOS(config)# admin adminuser1
```

2. To authorize external health monitor privileges, in addition to all read/write access, enter the `privilege hm` command.

```
ACOS(config-admin:adminuser1)# privilege hm
Modify Admin User successful!
```

3. Verify the modified `admin1`:

```
ACOS(config-admin:adminuser1)# show admin
```

UserName	Status	Privilege	Access	UserType
admin	Enabled	R/W/HM	C/W/A	Local
adminuser1	Enabled	R/W/HM	C/A	Local

External health-check scripts can be created, modified, and imported through CLI commands.

- Use the `health external create` to save a new health check script, then open an editor to modify that file.
- Use the `health external edit` command to open the editor to modify an existing script.
- Other CLI commands that manage health monitor scripts include `health external copy`, `health external delete`, and `health external rename`.

See the *Command Line Interface for ADC* for more information.

- Use the `import health-external` command at the global configuration level to import an external health-check script onto your ACOS device. The following example imports a script called `example-health.py`

```
ACOS(config)# import health-external example-health.py
ftp://examplehost/scripts/hc
```

The following example configures the external health monitor:

```
ACOS(config)# health monitor hm1
ACOS(config-health:monitor)# method external program example-health.py
```

- After the health monitor is configured, you can configure a server to use the external health method:

```
ACOS(config)# slb server rs1
ACOS(config-real server)# health-check hm1
```

See the *Command Line Interface* for more information about the Import command.

External Health Monitor Script Examples

The following topics are covered:

TCL Script Example	403
Perl Script Example	405
Shell Script Example	405
Python Script Example	406

TCL Script Example

For Tcl scripts, the health check parameters are transmitted to the script through the predefined TCL array ACOS_env. The array variable ax_env(ServerHost) is the server IP address and ax_env(ServerPort) is the server port number. Set ax_env(Result) 0 as pass and set the others as fail. TCL script filenames must use the “.tcl” extension.

To use the external method, you must import the program onto the ACOS device. The script execution result indicates the server status, which must be stored in ax_env (Result).

These commands import external program “ext.tcl” from FTP server 192.168.0.1, and configure external health method “hm3” to use the imported program to check the health of port 80 on the real server:

```
ACOS(config)# health external import "checking HTTP server"
ftp://192.168.0.1/ext.tcl
ACOS(config)# health monitor hm3
ACOS(config-health:monitor)# method external program ext.tcl port 80
```

Here is the ext.tcl file:

```
set sock -1
set ax_connected -1
set ax_result -1
set ax_server $ax_env(ServerHost)
set ax_port $ax_env(ServerPort)

proc recv_response { args } {
    global sock ax_result ax_server ax_port
    puts "recv_response"
    set line [read $sock]
    puts $line
    if { [ regexp "HTTP/1.. ([0-9\+]) " $line match status ] } {
        puts "server $ax_server:$ax_port response: $status"
        # Check the status code
        if { $status == 200 } {
            # Set server to be "UP"
            set ax_result 0
        } else {
            set ax_result 1
        }
    }
}
```

```
    } else {
        puts "not match"
        set ax_result 1
    }
    # clear the read event
    fileevent $sock readable {}
}

proc send_request { args } {
    global sock ax_result ax_connected
    puts "send_request"
    puts $sock "GET / HTTP/1.0\r\n"
    puts $sock "User-Agent: A10 HMON\r\n\r\n"
    set ax_connected 1
    # clear the write event
    fileevent $sock writable {}
}

# setup timer, 1000ms
after 1000 {
    set ax_connected 0
    set ax_result 1
    puts "timeout"
}

if {[catch {socket -async $ax_server $ax_port} sock]} {
    puts "$ax_server: $sock"
} else {
    fconfigure $sock -buffering none -blocking 0 -eofchar {}

    fileevent $sock writable { send_request 0 }
    puts "waiting"
    vwait ax_connected
    if {1 == $ax_connected} {
        fileevent $sock readable { recv_response 0 }
        vwait ax_result
    }
    set ax_env(Result) $ax_result
    close $sock
}
```


Perl Script Example

For other external scripts (non-Tcl), environment variables are used to pass the server IP address (HM_SRV_IPADDR) and the port number (HM_SRV_PORT). The script returns 0 as pass and returns others as fail. For Perl scripts, use `#!/usr/bin/perl` at the beginning of the script.

Here is an example using the Perl scripting language:

```
#!/usr/bin/perl -w
# Sample script for checking Web server

my $host = $ENV{'HM_SRV_IPADDR'};
my $port = 80;
if (defined($ENV{'HM_SRV_PORT'})) {
    $port = $ENV{'HM_SRV_PORT'};
}

# Use wget, exit code is zero if wget was successful.
my @args = qw(-O /dev/null -o /dev/null);
exec "wget", "http://$host:$port", @args;

# vim: tw=78:sw=3:tabstop=3:autoindent:expandtab
```

Shell Script Example

For Shell scripts, use `#!/bin/bash` at the beginning of the script.

The following example is the sample of using the Shell scripting language:

```
#!/bin/bash
if test "$HM_SRV_IPADDR" == "" ; then
    echo "Please check ENV Var 'HM_SRV_IPADDR'"
    exit 1
fi

echo -n "Test $HM_SRV_IPADDR ...."
wget $HM_SRV_IPADDR --delete-after --timeout=2 --tries=1 > /dev/null 2>&1

ret=$?
if test $ret == 0 ; then
    echo "OK"
```

```
exit 0
else
    echo "Fail"
exit 1
fi
```

Python Script Example

See [Database Health Monitoring Using a Script](#).

Database Health Monitoring Using a Script

ACOS supports Open Database Connectivity (ODBC), which enables Python-based external health monitors to perform Layer 7 health checks using information retrieved from a backend database.

The following types of databases are supported:

- Microsoft SQL (see [Example Script—Microsoft SQL](#))
- MySQL (see [Example Script—MySQL](#))
- PostgreSQL (see [Example Script—PostgreSQL](#))
- Oracle (see [Example Script—Oracle](#))

ACOS also provides a database health method. For more information, see [Database Health Monitors](#).

To configure database health monitoring using an external health monitor:

1. Configure a Python script to check the database. (See the examples below.)
2. Import the script onto the ACOS device.
3. Configure an external health monitor that uses the script.
4. Bind the external health monitor to the target (real server, real port, or service group).

The configuration workflow on the ACOS device is the same as that used for other external health monitor types.

Example Script—Microsoft SQL

```
#!/usr/bin/python
import sys
import os
import pyodbc

# MsSQL connection string format:
#Driver=FreeTDSDriver;Server=<ip>;Port=<port>;TDS_
version=7.0;Database=<database>;UID=<username>;PWD=<password>

# by al0hm convention get host and/or port from environment
host = os.environ['HM_SRV_IPADDR']
if os.environ.has_key('HM_SRV_PORT'):
    port = int(os.environ['HM_SRV_PORT'])
if 0 >= port or 65536 <= port:
    port = 0
else:
    port = 0

if 0 != port:
    conn_str = ("Driver=FreeTDSDriver;Server=%s;Port=%d;TDS_
version=7.0;Database=Database;UID=User;PWD=Password" % (host, port))
else:
    conn_str = ("Driver=FreeTDSDriver;Server=%s;TDS_
version=7.0;Database=Database;UID=User;PWD=Password" % (host))

sql_stmt = "select * from Table"
try:
    rv = 0
    print "Connecting %s" % (conn_str)
    conn = pyodbc.Connect(conn = conn_str)
    print "Doing SQL query '%s'" % (sql_stmt)
    conn.execute(sql_stmt)
    print "Fetching query results"
    rows = conn.fetch()
    print "Verifying results are OK"
    if (0 == len(rows)):
        rv = -1;
    print "Cleaning up the database connection"
    conn.disconnect()
```

```
del(conn)
except:
    print "Something went wrong"
    # by a10hm convention must exit with something other than 0
    rv = -1

sys.exit(rv)
```

Example Script—MySQL

```
#!/usr/bin/python
import sys
import os
import pyodbc

# MySQL connection string format:
#Driver=MySQLDriver;Server=<ip>;Port=<port>;Database=<database>;UID=<username>;PWD=<password>

# by a10hm convention get host and/or port from environment
host = os.environ['HM_SRV_IPADDR']
if os.environ.has_key('HM_SRV_PORT'):
    port = int(os.environ['HM_SRV_PORT'])
if 0 >= port or 65536 <= port:
    port = 0
else:
    port = 0

if 0 != port:
    conn_str =
("Driver=MySQLDriver;Server=%s;Port=%d;Database=Database;UID=User;PWD=Password" % (host, port))
else:
    conn_str =
("Driver=MySQLDriver;Server=%s;Database=Database;UID=User;PWD=Password" %
(host))

sql_stmt = "select * from Table"
try:
    rv = 0
```

```
print "Connecting %s" % (conn_str)
conn = pyodb.Connect(conn = conn_str)
print "Doing SQL query '%s'" % (sql_stmt)
conn.execute(sql_stmt)
print "Fetching query results"
rows = conn.fetch()
print "Verifying results are OK"
if (0 == len(rows)):
    rv = -1;
conn.disconnect()
del(conn)
except:
    print "Something went wrong"
    # by a10hm convention must exit with something other than 0
    rv = -1

sys.exit(rv)
```

Example Script—PostgreSQL

```
#!/usr/bin/python
import sys
import os
import pyodb

# PostgreSQL connection string format:
#
Driver=PostgreSQLDriver;Server=<ip>;Port=<port>;Database=<database>;UID=<u
sername>;PWD=<password>

# by a10hm convention get host and/or port from environment
host = os.environ['HM_SRV_IPADDR']
if os.environ.has_key('HM_SRV_PORT'):
    port = int(os.environ['HM_SRV_PORT'])
if 0 >= port or 65536 <= port:
    port = 0
else:
    port = 0

if 0 != port:
```

```
conn_str =
("Driver=PostgreSQLDriver;Server=%s;Port=%d;Database=Database;UID=User;PWD
=Password" % (host, port))
else:
    conn_str =
("Driver=PostgreSQLDriver;Server=%s;Database=Database;UID=User;PWD=Passwor
d" % (host))

sql_stmt = "select * from Table"
try:
    rv = 0
    print "Connecting %s" % (conn_str)
    conn = pyodbc.Connect(conn = conn_str)
    print "Doing SQL query '%s'" % (sql_stmt)
    conn.execute(sql_stmt)
    print "Fetching query results"
    rows = conn.fetch()
    print "Verifying results are OK"
    if ( 0 == len(rows)):
        rv = -1;
    print "Cleaning up the database connection"
    conn.disconnect()
    del(conn)
except:
    print "Something went wrong"
    # by al0hm convention must exit with something other than 0
    rv = -1

sys.exit(rv)
```

Example Script—Oracle

```
#!/usr/bin/python
import sys
import os
import pyodbc

# Oracle connection string format:
#Driver=OracleODBCDriver;DBQ=<Oracle-DBQ>;UID=<username>;PWD=<password>
```

```
# by a10hm convention get host and/or port from environment
host = os.environ['HM_SRV_IPADDR']
    if os.environ.has_key('HM_SRV_PORT'):
port = int(os.environ['HM_SRV_PORT'])
if 0 >= port or 65536 <= port:
    port = 0
else:
    port = 0

if 0 != port:
    conn_str =
("Driver=OracleODBCDriver;DBQ=//%s:%d/XE;UID=User;PWD=Password" % (host,
port))
else:
    conn_str = ("Driver=OracleODBCDriver;DBQ=//%s/XE;UID=User;PWD=Password"
% (host))

sql_stmt = "select * from Table"
try:
    rv = 0
    print "Connecting %s" % (conn_str)
    conn = pyodbc.Connect(conn = conn_str)
    print "Doing SQL query '%s'" % (sql_stmt)
    conn.execute(sql_stmt)
    print "Fetching query results"
    rows = conn.fetch()
    print "Verifying results are OK"
    if ( 0 == len(rows)):
        rv = -1;
    print "Cleaning up the database connection"
    conn.disconnect()
    del(conn)
except:
    print "Something went wrong"
    # by a10hm convention must exit with something other than 0
    rv = -1

# by a10hm convention must exit with 0
sys.exit(rv)
```

Alternate Real Servers and Ports for Backup

You can assign one or more backup servers as alternates for a given primary server. If that specific primary server goes down, one of its alternate servers takes over. Likewise, you also can assign alternates for backing up individual ports.

The following topics are covered:

Alternate Servers	413
Alternate Ports	417

Alternate Servers

You can assign up to 16 alternate servers to a primary server. Only 1 alternate server for a given primary server can be active at a time.

This feature places an alternate server into service *only* if the primary server goes down. Other features such as connection limiting or connection-rate limiting can not cause an alternate server to be used.

Sequence Numbering of Alternate Servers

Each alternate server must have a sequence number, 1-16. You specify the sequence number of an alternate server when you assign it to its primary server.

For example, the following set of servers consists of one primary server and 3 alternate servers:

- rs1 – Primary server
- rs1-a1 – Alternate server with sequence number 1
- rs1-a2 – Alternate server with sequence number 2
- rs1-a3 – Alternate server with sequence number 3

The alternate servers are used according to their sequence numbers, beginning with the lowest sequence number. For example, if all servers are up, then rs1 goes down, rs1-a1 takes over. If rs1-a1 also goes down, rs1-a2 takes over, and so on.

The sequence numbering of alternate servers is different from server priority. You do not need to change the priority values of the primary server or its alternate servers.

Re-Activation of Down Servers

When a down server comes back up, the server is placed back into service. New sessions can be sent to the server. The alternate server is gracefully removed from service. Existing sessions on the alternate server are allowed to continue until they are terminated or they time out.

Configuring Alternate Servers for Backup

To configure alternate servers for backup:

1. Add each of the alternate servers to the configuration.
2. Add the primary servers to the configuration. During configuration of each of the primary servers, specify the alternate servers to use as the primary server's backups.
3. Configure a service group. Add only the primary servers to the group. Do not add the alternate servers to the group.
4. Configure the virtual server. Bind the service group to a virtual port on the server.

The configuration options for the alternate servers are the same as those for primary servers. Likewise, the configuration options for the service group and virtual server are the same as those for configurations that do not use alternate servers.

The only server configuration that is unique to use of alternate servers is specification of those servers in the configuration of the primary servers.

GUI Configuration

To configure alternate servers for backup using the GUI:

1. Navigate to **ADC >> SLB >> Servers** tab.
2. Click the **Edit** link in the Actions column on the right side of the page for a primary server.
3. On the **Update Server** page, expand the Advanced Fields section.
4. In the Alternate Server field:
 - a. Enter the sequence number in the Server ID field.
 - b. Select an alternate server from the Server Name drop-down list.
 - c. Click **Add**.
 - d. Repeat for each alternate server to be used by the primary server as a backup.
5. Click **Update**.

CLI Configuration

To assign an alternate server as a dedicated backup for a primary server, use the **alternate** command at the configuration level for the primary server:

The following commands configure some real servers to be used as alternate servers for backup:

```
ACOS(config)# slb server rs1-a1 10.10.10.51
ACOS(config-real server)# port 80 tcp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
ACOS(config)# slb server rs1-a2 10.10.10.52
ACOS(config-real server)# port 80 tcp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
ACOS(config)# slb server rs1-a3 10.10.10.53
ACOS(config-real server)# port 80 tcp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
ACOS(config)# slb server rs2-a1 10.10.10.71
ACOS(config-real server)# port 80 tcp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
ACOS(config)# slb server rs2-a2 10.10.10.72
ACOS(config-real server)# port 80 tcp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
ACOS(config)# slb server rs2-a3 10.10.10.73
ACOS(config-real server)# port 80 tcp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
```

The following commands configure 2 primary servers, and assign alternate servers to each of them:

```
ACOS(config)# slb server rs1 10.10.10.10
ACOS(config-real server)# alternate 1 rs1-a1
ACOS(config-real server)# alternate 2 rs1-a2
ACOS(config-real server)# alternate 3 rs1-a3
ACOS(config-real server)# port 80 tcp
ACOS(config-real server-node port)# exit
```

```

ACOS(config-real server)# exit
ACOS(config)# slb server rs2 10.10.10.20
ACOS(config-real server)# alternate 1 rs2-a1
ACOS(config-real server)# alternate 2 rs2-a2
ACOS(config-real server)# alternate 3 rs2-a3
ACOS(config-real server)# port 80 tcp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit

```

The following commands configure a service group. Only the primary servers are added to the group. The alternate servers are not added to the group.

```

ACOS(config)# slb service-group http1 tcp
ACOS(config-slb svc group)# member rs1 80
ACOS(config-slb svc group-member:80)# exit
ACOS(config-slb svc group)# member rs2 80
ACOS(config-slb svc group-member:80)# exit
ACOS(config-slb svc group)# exit

```

The following commands configure a virtual server that uses the service group configured above:

```

ACOS(config)# slb virtual-server http-with-alternates 192.168.10.10
ACOS(config-slb vserver)# port 80 http
ACOS(config-slb vserver-vport)# service-group http1
ACOS(config-slb vserver-vport)# exit
ACOS(config-slb vserver)# exit

```

The following command indicates whether alternate servers are in use:

```

ACOS# show slb virtual-server bind
Total Number of Virtual Services configured: 1
-----
-----
*Virtual Server : http-with-alternates(A)                                192.168.10.10
Functional Up

    +port 80  http ==>http1                                           State :Functional
Up
    +rs1:80                                10.10.10.10                State : Up

    Alternate: rs1-a1, rs1-a2, rs1-a3
    +rs2:80                                10.10.10.20                State : Down

```

```
Alternate: rs2-a1*, rs2-a2, rs2-a3
```

Primary servers are listed under the virtual port. Under each primary server, their alternate servers are listed.

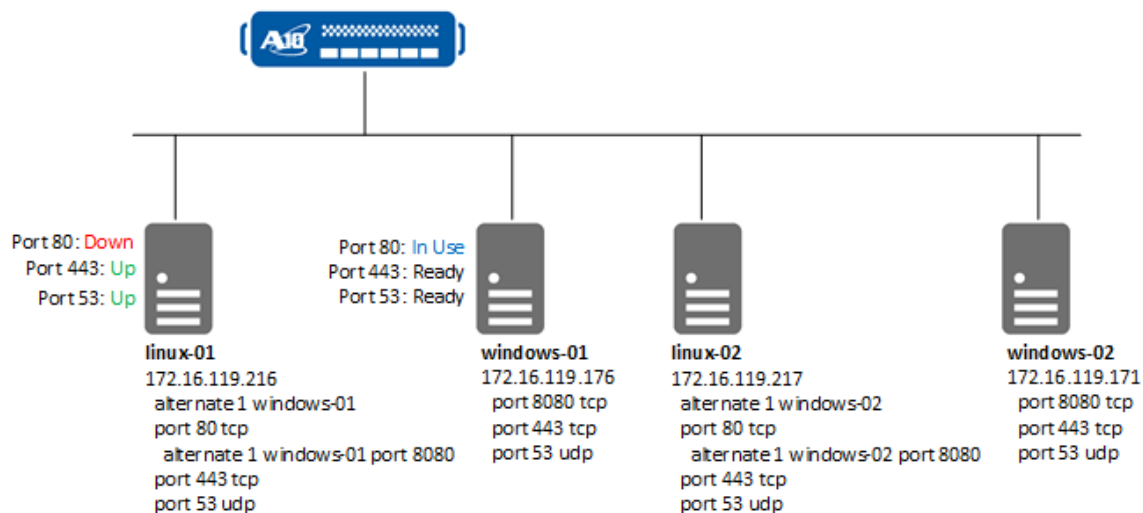
If an asterisk is shown at the end of an alternate server name, the primary server is down and the alternate server is active instead. In the example above, rs2 is down, so alternate rs2-a1 is being used instead.

Alternate Ports

Alternate Ports Overview

For more granularity, you can specify alternates for individual ports. In this case, if a primary port that has an alternate goes down, the ACOS device uses the alternate for that port, but continues to use the primary server for other ports, if those other ports are still up. [Figure 45](#) shows an example.

Figure 45 : Alternate Server Ports



This deployment uses two primary servers, “linux-01” and “linux-02”. An alternate server is configured for each of the primary servers. The server “windows-01” is an alternate server for “linux-01”. Likewise, “windows-02” is an alternate server for “linux-02”.

Each primary server, and each of its alternate servers, has the following ports:

- TCP port 80 (or 8080)
- TCP port 443
- UDP port 53

As this example shows, port numbers on the primary and alternate servers are not required to be the same. TCP port 8080 on the alternate servers provides the backup for port 80 on the primary servers.

Although port 80 has its own alternate port, the deployment in this example does not use alternate ports for 443 or 53. If port 443 or 53 on a primary server goes down, the ACOS device starts using the alternate server, for all the primary server's ports (443 and 53).

However, if only port 80 goes down, the ACOS device uses port 8080 on the alternate server but continues using the primary server for ports 443 and 53. This is because port 80 has its own alternate port.

For simplicity, this example shows only a single alternate server for each primary server. In fact, a primary server can have up to 16 alternate servers. In either case, for a given primary server, only one alternate server can be in use at a time. Likewise, a port can have up to 16 alternate ports but only one of those ports can be in use at any given time. Priority ranges from highest (1) to lowest (16).

For information about how the alternate-server feature works, see [Alternate Servers](#).

Configuring Alternate Ports

To configure alternate ports:

1. Configure an alternate server.
2. Configure the primary server.
 - a. Add the alternate server to the primary server.
 - b. Add the protocol ports. On each port for which you want to provide an individual backup, add the alternate server and port.

Use the same sequence number for the alternate server and the alternate port. This is required.

3. Configure the service group.

Within the service group configuration, there is no specific configuration required for the alternate servers and ports. Add only the primary server and ports as members of the group. Do not add the alternate servers or ports to the service group. No service group is required for the alternate servers and ports.

4. Configure the virtual server. Bind the primary server's service group to the virtual port. In the virtual server configuration, there is no specific configuration required for the alternate servers and ports.

NOTE:

- The alternate port sequence number be identical to the alternate server sequence number.
 - A given alternate server can be used by only one primary server within the same service group.
-

GUI Configuration

To configure alternate ports for backup using the GUI:

1. Navigate to **ADC >> SLB >> Servers** tab.
2. Click the **Edit** link in the Actions column on the right side of the page for a primary server.
3. In the Port section on the Update Server page, click the **Edit** link for a primary port.
4. On the Update Port page, expand the Advanced Fields section.
5. In the Alternate Port field:
 - a. Select the alternate server from the Server ID/Name drop-down list.
 - b. Select an alternate port from the Server Port drop-down list.
 - c. Click **Add**.

d. Repeat for each alternate port to be used by the primary port as a backup.

6. Click **Update**.

CLI Configuration

To configure alternate ports, use the **alternate** command at the configuration level for the real server.

The commands in this example configure the deployment shown in [Alternate Server Ports](#).

To begin, the following commands configure the alternate servers:

```
ACOS(config)# slb server windows-01 172.16.119.176
ACOS(config-real server)# port 8080 tcp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# port 443 tcp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# port 53 udp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
ACOS(config)# slb server windows-02 172.16.119.171
ACOS(config-real server)# port 8080 tcp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# port 443 tcp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# port 53 udp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
```

The following commands configure the primary servers:

```
ACOS(config)# slb server linux-01 172.16.119.216
ACOS(config-real server)# alternate 1 windows-01
ACOS(config-real server)# port 80 tcp
ACOS(config-real server-node port)# alternate 1 windows-01 port 8080
ACOS(config-real server-node port)# exit
ACOS(config-real server)# port 443 tcp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# port 53 udp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
```



```
ACOS(config)# slb server linux-02 172.16.119.217
ACOS(config-real server)# alternate 2 windows-02
ACOS(config-real server)# port 80 tcp
ACOS(config-real server-node port)# alternate 2 windows-02 port 8080
ACOS(config-real server-node port)# exit
ACOS(config-real server)# port 443 tcp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# port 53 udp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
```

The fact that these servers are not used as alternates for other servers makes these “primary servers”.

These commands configure the service groups. A separate service group is configured for each service:

```
ACOS(config)# slb service-group linux-http tcp
ACOS(config-slb svc group)# member linux-01 80
ACOS(config-slb svc group-member:80)# exit
ACOS(config-slb svc group)# member linux-02 80
ACOS(config-slb svc group-member:80)# exit
ACOS(config-slb svc group)# exit
ACOS(config)# slb service-group linux-ssl tcp
ACOS(config-slb svc group)# member linux-01 443
ACOS(config-slb svc group-member:80)# exit
ACOS(config-slb svc group)# member linux-02 443
ACOS(config-slb svc group-member:80)# exit
ACOS(config-slb svc group)# exit
ACOS(config)# slb service-group linux-dns udp
ACOS(config-slb svc group)# member linux-01 53
ACOS(config-slb svc group-member:80)# exit
ACOS(config-slb svc group)# member linux-02 53
ACOS(config-slb svc group-member:80)# exit
ACOS(config-slb svc group)# exit
```

The following commands configure the virtual server:

```
ACOS(config)# slb virtual-server vip1 192.168.19.120
ACOS(config-slb vserver)# port 80 http
ACOS(config-slb vserver-vport)# service-group linux-http
ACOS(config-slb vserver-vport)# exit
```

```

ACOS(config-slb vserver)# port 443 https
ACOS(config-slb vserver-vport)# service-group linux-ssl
ACOS(config-slb vserver-vport)# exit
ACOS(config-slb vserver)# port 53 dns-udp
ACOS(config-slb vserver-vport)# service-group linux-dns
ACOS(config-slb vserver-vport)# exit
ACOS(config-slb vserver)# exit

```

The following command displays binding information for the virtual server. This information includes the status of the primary and alternate servers and ports, for the service-group members bound to the virtual port.

```

ACOS# show slb virtual-server bind
Total Number of Virtual Services configured: 3
-----
----
*Virtual Server : vip1(A)                192.168.19.120  Functionally
Up
  +port 80  http ==>linux-http                State :
Functionally Up
    +linux-01:80                172.16.119.216      State : Down
    +linux-02:80                172.16.119.217      State : Up
  +port 443  https ==>linux-ssl                State :Up
    +linux-01:443                172.16.119.216      State : Up
Alternate: windows-01
    +linux-02:443                172.16.119.217      State : Up
Alternate: windows-02
  +port 53  dns-udp ==>linux-dns                State :Up
    +linux-01:53                172.16.119.216      State : Up
Alternate: windows-01
    +linux-02:53                172.16.119.217      State : Up
Alternate: windows-02

```

The output indicates that port 80 on “linux-80” is Down. Therefore, alternate port 8080 on server “windows-01” is used instead.

Alternate Virtual Ports for Backup

The following topics are covered:

Overview	424
Configuring Alternate Virtual Ports for Backup	425

Overview

ACOS supports use of backup virtual ports to provide backup for other virtual ports. With this feature enabled, ACOS can switch over to a different virtual port under certain events. In this way, the feature is similar to the Alternate Ports feature (see [Alternate Real Servers and Ports for Backup](#).)

The following topics are covered:

Supported Alternate Virtual Port Service Types	424
Virtual Port Switchover Options	424

Supported Alternate Virtual Port Service Types

The alternate virtual port feature applies to the following service types:

- TCP
- HTTP

The feature offers the ability to take advantage of ACOS' high performance Layer 4 load balancing while offering the option to allow different service groups to handle different types of traffic, to add an alternate service group for backup purposes, or to simply return an error message to clients. For example, the alternate virtual port feature could be used to send clients an error message, such as "page not found", using an aFlex script, or it could be used to trigger a backup service group to handle this request.

Virtual Port Switchover Options

The following events could cause the primary virtual port to switchover to an alternate virtual port:

- when-down – This option applies to Layer 4 or Layer 7 virtual ports. You can configure an alternate virtual port for use as a backup, and client requests will be re-directed if the primary virtual port is down.

- **serv-sel-fail** – This option applies to only Layer 4 primary virtual ports. The server selection switchover could occur for any number of reasons that could cause server selection to fail, such as if the service group configured on the primary virtual port reaches a configured connection limit.
- **req-fail** – This option only applies if the primary virtual port is HTTP. If a non-HTTP TCP request is sent to an HTTP virtual port, then the client request will switchover to an alternate virtual port with service type TCP. For example, in some cases it may be desirable to have the ACOS device load balancing non-HTTP traffic using a Layer 4 service group.

This req-fail option only works for the first request in the connection.

Configuring Alternate Virtual Ports for Backup

To configure the alternate virtual ports for backup feature:

1. Set up the real servers and service groups.
2. When setting up the virtual server (VIP), configure a primary virtual port with TCP or HTTP.
3. Add an alternate virtual port. The alternate virtual port and primary virtual port must be of different service types. For example:
 - An HTTP alternate virtual port must be configured with a TCP primary virtual port

or

 - An TCP alternate virtual port must be configured with an HTTP primary virtual port.

Configuring Alternate Virtual Ports (GUI Example)

This example configures virtual server vs1 with TCP port 80 as the primary virtual port, and HTTP port 81 as the alternate virtual port.

1. Select **ADC > SLB**.
2. Select the **Virtual Services** tab.
3. On the Create Virtual Service page:

- a. Enter **vs1** in the Virtual Server Name field.
 - b. Enter **10.10.10.1** in the Address field.
 - c. Select **TCP** from the drop-down list in the Protocol field.
 - d. Enter **80** in the Port field.
4. Expand the General Fields section of the page.
5. Select the checkbox in the Use Alternate Port field.
 - a. Enter **81** in the Alternate Port field.
 - b. Select **HTTP** in the Alternate Protocol field.
 - c. Select the desired condition in the Alternate Condition field.
 - d. Click **Create**.

Configuring Alternate Virtual Ports (CLI Example)

This example assumes the real servers have already been configured, and that two service groups have been configured: one for the primary virtual port and another for the alternate virtual port.

The VIP is configured with an HTTP primary virtual port and a TCP alternate virtual port, and the event trigger used in this example is 'req-fail'.

```
ACOS(config)# slb virtual-server altvip1 5.5.5.7
ACOS(config-slb vserver)# port 80 tcp alternate
ACOS(config-slb vserver-vport)# service-group sg-80-tcp-alt
ACOS(config-slb vserver-vport)# exit
ACOS(config-slb vserver)# port 80 http
ACOS(config-slb vserver-vport)# alternate port 80 tcp req-fail
ACOS(config-slb vserver-vport)# service-group sg-80-tcp-prim
```

Upon sending non-HTTP TCP traffic to the HTTP virtual port, the traffic should trigger a failed client request (based on the inconsistent client request and service type of the virtual port). This triggers a failover (or switchover) to the TCP alternate port, and this virtual port belonging to service group "sg-80-tcp-alt" should be able to accommodate the TCP traffic.

Priority Affinity

Priority affinity is a service-group option that allows the ACOS device to continue using backup servers (servers with lower priority) even when the primary (high priority) servers come back up.

The following topics are covered:

Priority Affinity Example	428
Priority Affinity Options	430
Configuring Priority Affinity	432

Priority Affinity Example

The following example helps illustrate how priority affinity works.

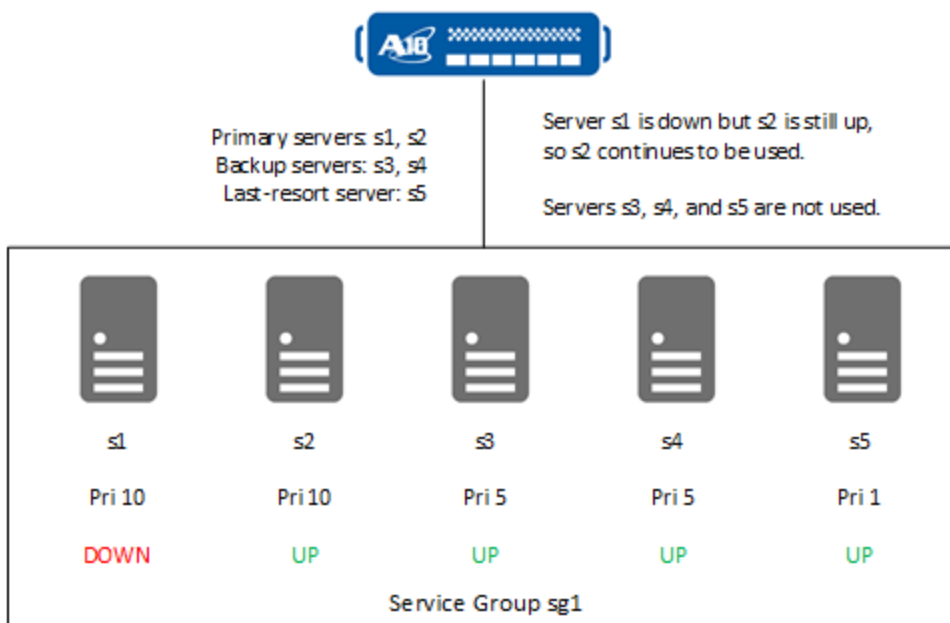
Assume a service group contains five members, as shown below:

- Server s1, port 80, priority 10
- Server s2, port 80, priority 10
- Server s3, port 80, priority 5
- Server s4, port 80, priority 5
- Server s5, port 80, priority 1

All five members of this service group (servers s1-s5) are up and available. However, the ACOS device uses only members s1 and s2, because they have a priority of 10. Members s3 and s4 are used only if both s1 and s2 go down. Member s5 is a last-resort member that is used only if all other members are down.

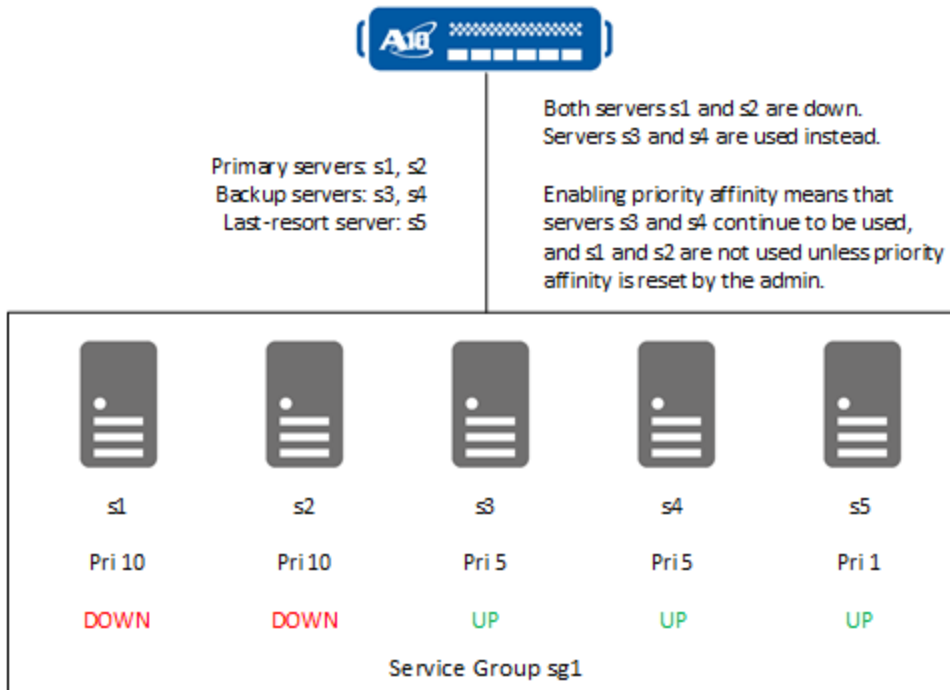
If server s1 goes down, as shown in [Figure 46](#), the ACOS device continues sending traffic to the other highest-priority server, s2.

Figure 46 : Priority Affinity – First Server Fails



Next, assume server s2 also goes down, as shown in [Figure 47](#) below. Because s1 and s2 are both down, the ACOS device begins using the backup servers (s3 and s4).

Figure 47 : Priority Affinity – Second Server Fails



By default, without priority affinity, if s1 or s2 returns to an available state, the ACOS device stops using s3 and s4 and shifts back to s1 and s2. However, with priority affinity enabled, the ACOS device continues to use s3 and s4 and does not start using s1 or s2 again, despite their availability.

If the ACOS continues to use the lower-priority servers and you wish to force the ACOS to use the higher-priority servers again, you must administratively reset the priority affinity.

ACOS Behavior – Priority Affinity Disabled

By default, the priority affinity feature is disabled when a service group is created. With priority affinity disabled, the ACOS device's default behavior is such that:

1. ACOS sends traffic to the service-group members with the highest priority.
2. If these high-priority servers go down, then the ACOS device selects the service-group members with the next-highest priority.
3. However, if one or more of the high-priority servers return to service, ACOS stops sending traffic to the low-priority servers and reselects the high-priority servers.

ACOS Behavior – Priority Affinity Enabled

If the priority affinity feature is enabled for the service group, the behavior of the ACOS device continues sending clients to lower-priority servers, even after the higher-priority servers have come back online.

Previous releases provided an option (min-active-member) that enables the ACOS to continue using backup servers. This approach can ensure availability of a configured minimum number of active servers but, unlike priority affinity, the min-active-member option lacks a way to ensure traffic is only sent to backup servers.

If the ACOS device stops using the primary servers due to other features (for example, exceeding connection limits), then the priority affinity feature will take effect just as if the switchover were triggered by a change in the status of the primary servers. If the higher-priority servers subsequently become available due to the number of connections dropping below the configured threshold, then the ACOS device will not use them, but will instead continue using the lower-priority backup servers.

Priority Affinity Options

Within the priority affinity feature, there are sub-options that enable you to define specific actions that will occur if the higher-priority service-group members fail.

This ability to specify a particular action to occur during a failover may be helpful because it allows you to prevent your lower-priority secondary servers, (which are presumably less robust than your higher-priority servers), from being overwhelmed by a flood of traffic that could occur during failover.

Actions (drop, reset, and others) can be tied to a general failure (such as disabled service-group members or a failed health check) or to traffic exceeding configured connection-limits or connection-rate-limits.

Priority Affinity Actions

Configure the ACOS device to respond to service-group member failures with one of the following actions:

- **Reset:** Sends a reset to the client if all nodes with this same priority fail for any reason
- **Reset-if-exceed-limit:** Sends a reset to the client if all nodes with this same priority fail, and if failure is due to one or more nodes exceeding the configured connection-limit or connection-rate-limit
- **Drop:** Drops the request if all nodes with this same priority fail for any reason
- **Drop-if-exceed-limit:** Drops the request if all nodes with this same priority fail, and if one or more nodes exceed the configured connection-limit or connection-rate-limit
- **Proceed:** the ACOS device uses the node(s) with the next-highest priority if all nodes with the currently-selected priority fail (this is the default behavior)

Actions are tied to a certain priority levels and are *not* tied to individual servers. Therefore, an action is only triggered when *all* service-group members of a given priority become unavailable.

Priority Affinity Triggers

The reset or drop actions can be triggered for the following reasons:

- If a health check fails
- If a user disables a server or port
- If another Load Balancing feature causes the currently-used priority to become unavailable (for example, min-active-member)
- If a connection-limit or connection-rate-limit is exceeded

Configuring Priority Affinity

The following topics are covered:

Configuring Basic Priority Affinity	432
Configuring Affinity Priority Levels	433
Configuring ACOS Reactions to Exceeded Connection Limits	434

Configuring Basic Priority Affinity

GUI Configuration

1. Navigate to **ADC >> SLB >> Service Group** tab.
2. Click the **Edit** link in the Actions column for the service group you want to modify.
3. Expand the Advanced Fields section.
4. Click the checkbox in the Priority Affinity field.
5. Click **Update**.

CLI Configuration

The following commands add members to a service group. Members s1 and s2 have the highest priority value within the group, so they will be used as the primary members. Members s3 and s4 will be used only if both s1 and s2 become unavailable. Member s5 will be used only if all the other members are unavailable.

```
ACOS(config)# slb service-group sg1 tcp
ACOS(config-slb svc group)# member s1 80
ACOS(config-slb svc group-member:80)# priority 10
ACOS(config-slb svc group-member:80)# exit
ACOS(config-slb svc group)# member s2 80
ACOS(config-slb svc group-member:80)# priority 10
ACOS(config-slb svc group-member:80)# exit
ACOS(config-slb svc group)# member s3 80
ACOS(config-slb svc group-member:80)# priority 5
```

```
ACOS(config-slb svc group-member:80)# exit
ACOS(config-slb svc group)# member s4 80
ACOS(config-slb svc group-member:80)# priority 5
ACOS(config-slb svc group-member:80)# exit
ACOS(config-slb svc group)# member s5 80
ACOS(config-slb svc group-member:80)# priority 1
ACOS(config-slb svc group-member:80)# exit
ACOS(config-slb svc group)# priority-affinity
ACOS(config-slb svc group)# exit
```

The **priority-affinity** command enables priority affinity; if the primary members both become unavailable, the secondary members (s3 and s4) will continue to be used even if s1 or s2 becomes available again.

In this example, primary members (the ones with highest priority within the service group) are active, so the priority affinity value is the priority of those members. However, if the primary members are down and the secondary members are active, the priority affinity value changes to the priority of the secondary members.

```
ACOS(config-slb svc group)# show slb service-group sgl
Service group name: sgl          State: All Up
Service selection fail drop:     498
Service selection fail reset:    0
Service peak connection:        0
Priority affinity:                5
...
```

If the output indicates that the priority affinity value is 0, this indicates that none of the service group's members have ever had any active SLB sessions. Typically, 0 appears when the service group is new and has not yet received any SLB traffic.

Configuring Affinity Priority Levels

Associating Actions with Priority Levels (CLI Example)

These commands create the TCP service-group "http" and defines the **reset-if-exceed-limit** action for members with priority 10. They also define the **drop-if-exceed-limit** action for members with priority 8.

```
ACOS(config)# slb service-group http tcp
ACOS(config-slb svc group)# priority 10 reset-if-exceed-limit
```

```

ACOS(config-slb svc group)# priority 8 drop-if-exceed-limit
ACOS(config-slb svc group)# member no30 80
ACOS(config-slb svc group-member:80)# priority 10
ACOS(config-slb svc group-member:80)# exit
ACOS(config-slb svc group)# member no31 80
ACOS(config-slb svc group-member:80)# priority 8
ACOS(config-slb svc group-member:80)# exit
ACOS(config-slb svc group)# member no32 80
ACOS(config-slb svc group-member:80)# priority 5
ACOS(config-slb svc group-member:80)# exit
ACOS(config-slb svc group)# member no33 80
ACOS(config-slb svc group-member:80)# exit
ACOS(config-slb svc group)# member no34 80
ACOS(config-slb svc group-member:80)# exit
ACOS(config-slb svc group)# exit

```

Use the **show slb service-group** command to display information about the service-group “http” created above. Output shows there were 23 packets dropped and 41 connections reset:

```

ACOS(config)# show slb service-group http
Service group name: http                               State: Functional Up
Service selection fail drop:      23
Service selection fail reset:    41
Service peak connection:         0
Service: no30:80                                     DOWN
Forward packets:      0      Reverse packets:      0
Forward bytes:        0      Reverse bytes:        0
Current connections:  0      Persistent connections: 0
Current requests:     0      Total requests:      0
Total connections:   0      Response time:      0    tick
Total requests succ:  0
Peak conn:           0

```

Configuring ACOS Reactions to Exceeded Connection Limits

The following examples show scenarios in which the ACOS device performs a certain action based on general failures or exceeded connection limits or exceeded connection rate limits:

- [Service Group with Two Priorities \(CLI Example\)](#)
- [Multiple Members with Same Priority \(CLI Example\)](#)
- [Different Actions for Different Priority Levels \(CLI Example\)](#)

Service Group with Two Priorities (CLI Example)

Consider the service-group below, which has two priorities (10 and 5). The **reset-if-exceed-limit** or **drop-if-exceed-limit** command can be applied to the higher priority level in order to reset the connection if the connection limit is exceeded.

```
ACOS(config)# slb service-group sgl tcp
ACOS(config-slb svc group)# priority 10 reset-if-exceed-limit
ACOS(config-slb svc group)# member A 80
ACOS(config-slb svc group-member:80)# priority 10
ACOS(config-slb svc group-member:80)# exit
ACOS(config-slb svc group)# member B 80
ACOS(config-slb svc group-member:80)# priority 5
ACOS(config-slb svc group-member:80)# exit
ACOS(config-slb svc group)# exit
```

With this configuration, if member A (priority 10) exceeds the configured connection-limit, the ACOS device responds by sending a reset to the client.

This reset only happens if the connection limit is exceeded. If member A goes down for a different reason, such as failing a health check, then the ACOS device will not perform the specified action. Instead, the ACOS device will act according to its default behavior, meaning it will reselect the server within the service-group that has the next-highest priority. In this example, this means member B (priority of 5), would be used.

Multiple Members with Same Priority (CLI Example)

Assume the service-group has several members with same priority level.

```
ACOS(config)# slb service-group sgl tcp
ACOS(config-slb svc group)# priority 10 reset-if-exceed-limit
ACOS(config-slb svc group)# member A 80
ACOS(config-slb svc group-member:80)# priority 10
ACOS(config-slb svc group-member:80)# exit
ACOS(config-slb svc group)# member B 80
ACOS(config-slb svc group-member:80)# priority 10
```

```
ACOS(config-slb svc group-member:80)# exit
ACOS(config-slb svc group)# member C 80
ACOS(config-slb svc group-member:80)# priority 10
ACOS(config-slb svc group-member:80)# exit
ACOS(config-slb svc group)# member D 80
ACOS(config-slb svc group-member:80)# priority 5
ACOS(config-slb svc group-member:80)# exit
ACOS(config-slb svc group)# exit
```

In this case, members A, B, and C all have a priority of 10. The specified action (**reset-if-exceed-limit**) only occurs if all three high-priority members fail and one or more of the failures is caused by exceeded connection limit. If members A, B, and C go down for any other reason, such as a failed health check, then the ACOS device would follow its default behavior and send traffic to the lower-priority service-group member D.

Different Actions for Different Priority Levels (CLI Example)

You can define different actions for different priority levels. Members A, B, C, and D in the next example have different priority levels. Different actions are associated with each of the different priority levels.

```
ACOS(config)# slb service-group sg1 tcp
ACOS(config-slb svc group)# priority 10 reset-if-exceed-limit
ACOS(config-slb svc group)# priority 8 drop-if-exceed-limit
ACOS(config-slb svc group)# priority 5 reset-if-exceed-limit
ACOS(config-slb svc group)# member A 80
ACOS(config-slb svc group-member:80)# priority 10
ACOS(config-slb svc group-member:80)# exit
ACOS(config-slb svc group)# member B 80
ACOS(config-slb svc group-member:80)# priority 8
ACOS(config-slb svc group-member:80)# exit
ACOS(config-slb svc group)# member C 80
ACOS(config-slb svc group-member:80)# priority 5
ACOS(config-slb svc group-member:80)# exit
ACOS(config-slb svc group)# member D 80
ACOS(config-slb svc group-member:80)# priority 1
ACOS(config-slb svc group-member:80)# exit
ACOS(config-slb svc group)# exit
```


- The Priority Options feature operates at Layer 4 feature and thus will not work if applied to virtual-ports, such as HTTP and SMTP, which are Layer 7. A10 suggests you use L4 virtual-ports.
- The Priority Options feature is only supported for the default service-group binding under the virtual-port. If the service-group has been configured under aFlex, or a black/white list, or a class list, then the specified action (e.g. drop, reset, proceed) will not take effect.
- Rate limiting and priority are not supported with stateless load balancing. Therefore, the Priority Options feature is also not supported in stateless load balancing implementations.

Source-IP Persistence Override and Reselect

When the Source-IP Persistence Override and Reselect feature is enabled, the ACOS device checks for higher-priority servers even if persistent sessions are already established between client and server.

The following topics are covered:

- [Default Source IP Persistence Behavior](#) 439
- [Behavior With Source-IP Persistence Override and Reselect](#) 439
- [Simplified Source IP Persistence Override Example](#) 439
- [Source IP Persistence Override Use Case Scenario](#) 440
- [Configuring Source IP Persistence Override](#) 440

Default Source IP Persistence Behavior

Without this feature, if source-IP persistence is enabled and if a persistent session has already been established between a client and a server, then the ACOS device will send traffic to that same node until the node goes down or the timeout period expires.

This “sticky” behavior (or persistence) is helpful in situations where it is important to maintain a consistent connection between a client and a server, such as with online shopping, where it may be desirable to track the client’s interaction with the website.

ACOS typically uses the priority metric to select the highest priority server from a service group, and it establishes a persistent session between the client and the selected server. Once the session is established, the server selection process is complete, and the priority of one server over another becomes irrelevant. Even if a higher-priority server becomes available, the ACOS device will “ignore” it and continue to honor the persistent session that has already been established.

Behavior With Source-IP Persistence Override and Reselect

If the Source-IP Persistence Override and Reselect feature is enabled, then the ACOS device no longer honors the source-IP persistence, and it continually checks for higher-priority servers, even after persistent sessions have been established between client and server.

Simplified Source IP Persistence Override Example

For example, assume a service group has two servers and traffic is load balanced across the two servers with persistence:

- Server A with priority = 10
- Server B with priority = 5

If Server A goes down, then the traffic is balanced to Server B, which has a lower priority. A persistent connection is established with Server B.

However, because the *Source-IP Persistence Override and Reselect* feature is enabled, when Server A comes back up, the persistence with Server B is broken and a new persistent session is re-established with Server A.

Source IP Persistence Override Use Case Scenario

The behavior described above might be desirable if you want to send clients to higher-priority servers as they become available. For example, this feature might be helpful if you have a large service group containing your newest and most robust servers, as well as a smaller service group containing a few older and weaker backup servers that have a lower-priority.

If you are doing off-hours maintenance on the high-priority servers, you must take them offline. Traffic will be sent to the low-priority servers in the other service-group.

However, once the maintenance is complete and you bring the high-priority servers back online, you might want to interrupt the persistent sessions that clients have established with your inferior backup servers. These persistent sessions will be broken in order to re-establish persistent sessions with your more robust, high-priority servers.

Configuring Source IP Persistence Override

GUI Configuration

To configure source-IP persistence override using the GUI:

1. Navigate to **ADC >> Templates >> Persistence** tab.
2. Click **Create**, then select **Persist Source IP** from the pop-up menu.
3. Enter a name for the template.
4. Select the checkbox in the Enforce Higher Priority field.
5. Click **OK**.

CLI Configuration

To enable source-IP persistence override and reselect, configure an SLB template for source-IP persistence and then use the **enforce-higher-priority** command. This example shows commands required to configure this feature and does not represent a complete SLB configuration.

The following command creates a source-IP persistence template named “SIP” and enables source-IP persistence override and reselect:

```
ACOS(config)# slb template persist source-ip sip
ACOS(config-source ip persist)# enforce-higher-priority
ACOS(config-source ip persist)# exit
```

The following commands create a virtual-server named “VIP1” at IP address 1.2.3.4 on TCP port 80. The service-group “HTTP” and source-IP persistence template “SIP” are then bound to this virtual server.

```
ACOS(config)# slb virtual-server vip1 1.2.3.4
ACOS(config-slb vserver)# port 80 tcp
ACOS(config-slb vserver-vport)# service-group http
ACOS(config-slb vserver-vport)# template persist source-ip sip
ACOS(config-slb vserver-vport)# exit
ACOS(config-slb vserver)# exit
```

Use the **show slb persist** command to display information about the status of the source-IP persistence override and reselect. The very last line in the output below shows that the ACOS “broke” 30 persistent sessions between a client and a low-priority node. This means server reselection occurred and new persistent sessions were established with higher-priority nodes.

```
ACOS(config)# show slb persist
Total
-----
URL hash persist (pri)      0
URL hash persist (sec)     0
...
Cookie persist ok          0
Cookie persist fail        0
Persist cookie not found   0
Persist cookie Pass-thru   0
Enforce higher priority    30
```

Policy-Based (PBSLB) Templates

This section describes various options and configuration examples to apply PBSLB templates.

The following topics are covered:

PBSLB Options	443
Forward Policy Options	444
Binding Class-List to PBSLB Template	452
Binding PBLSB Templates to Service Groups	454
Binding Policy Template to Virtual Ports	456
Viewing PBSLB Statistics	460

PBSLB Options

You can configure the following options in the PBSLB template and bind them to either service-group or virtual-ports.

- Class List - Specifies a system-wide class-list or PBSLB class-list or geo-location class-list within the template.
- [Forward-policy](#) - Configures a forward policy of PBSLB template to specify permitted traffic destinations and sources along with the actions to apply.
- Forward Proxy End-to-end HTTP/2 Support - Enables end-to-end HTTP/2 support for forward-proxy deployments, including proxy-chaining scenarios. This option preserves HTTP/2 connections on both the client and backend sides. By default, this option is disabled.
- Geo-location - Configures ACOS to prefer sites that are geographically closer to the clients. It is recommended to enable or disable this option before enabling GSLB. Changing the state of this option while GSLB is running can cause the related statistics counters to be incorrect.
 - Full Domain Tree - Checks current connection count for the client's specific geo-location and for all geo-locations higher up in the domain tree.
 - OverLap - Enables overlap matching mode. If there are overlapping addresses in the class-list, this option enables ACOS device to find the most precise match.
 - Share - Enables sharing of PBSLB statistics counters for virtual servers and virtual ports that use the template.
- Sampling Enable - Enables baseline for PBSLB traffic.
- User Tag - Assigns a user-defined tag to the PBSLB template for identification and management purposes.

Apart from the above configuration options for PBSLB, you can also globally set the PBSLB entries age timeout under SLB common configuration. The default age is set to 6 minutes.

```
ACOS(config)# slb common
ACOS(config-common)# pbslb-entry-age 2
```

NOTE: If the PBSLB entries current age is less than the set value, then it is updated to the new value on the next traffic. If the PBSLB entries current age is greater than the set value, then it will remain at that value.

Forward Policy Options

The forward-proxy solution of ACOS is available in the 'forward-policy' of the PBSLB template. It offers a variety of match rules and conditions to facilitate more granularity and control over which client traffic is forwarded or bypassed to the servers. You can implement ACLs, source and destination rules, match conditions, actions to be taken for each rule, and logging to analyze attempted policy violations. This solution also provides two configuration modes:

- Standard (legacy) mode - To configure source and destination rules with only one match condition.
- Advanced mode - To configure fine-grained access control using multiple source or destination conditions and actions.

For creating a forward policy, use the `forward-policy` sub-command mode and implement the following options, rules, and match conditions as per the business requirement:

- Source Rule (`source`) – Defines the matching rule for the traffic sources. The source sub-command within the forward policy names the source rule. Multiple source rules may be defined in a forward policy. But only a single source rule of match-any may be defined for a forward policy. IP addresses configured for a forward policy in multiple source class-lists cannot overlap.
 - Match class-list (`match-class-list`) – IPv4 or IPv6 class list that specifies hosts or subnets for matching source rules.
 - Match any (`match-any`) - Default match rule if no other specific class list is matched.
 - Match authorize policy (`match-authorize-policy`) - Authorize-policy for user and group-based policy.

- Priority (**priority**)- Priority of the source (higher the number, higher the priority, default 0).
- Destination Rule (**destination**) – The destination match rule is associated with a source rule and defines the following options.
 - Advance Match (**adv-match**) - Advance match rule options. Make sure to **enable-adv-match** option.
 - Match options - Configure various match conditions to filter the HTTP requests and enable drop or bypass action. You can also [configure forward proxy objects](#) as an additional action or condition.
 - Disable REQMOD or RESPMOD ICAP templates (**disable-reqmod-icap** | **disable-respmod-icap**) - Disable REQMOD and/or RESPMOD ICAP templates bound to the virtual port when the HTTP request matches the 'destination' advance match rule.
 - Any (**any**) - Default destination match rule if no other specific class list is matched.
 - Class-List (**class-list**) – A class list containing the URL, hostname strings, or IP range for destinations that clients can access.
 - Web-Category-List (**web-category-list**) - Category list containing predefined website URLs for destinations that clients can access. The web-category license file must be imported, and then **enable** must be invoked before using the category-list within the web-category sub-configuration.
 - Web-reputation-scope (**web-reputation-scope**) - Reputation scope containing scope name for destinations that clients can access.
- Enable Advanced Match (**enable-adv-match**) - Enables advanced mode to configure advanced match conditions. These conditions are available only in the **destination adv-match** rule and apply to HTTP or HTTPS virtual ports.

NOTE: Binding or switching the adv-match mode forward-policy template on a virtual port will enable **support-http2** implicitly.

- Action list (**action**)– Defines how traffic will be forwarded. These are mutually exclusive actions; multiple actions may be configured under a single policy. The following actions can be configured:

- Drop message (**drop-message**) - Drop message sent to the client as a webpage (HTML tags are included, and quotation marks are required for white spaces).
- Drop redirect URL (**drop-redirect-url**) - URL to which a dropped client request is redirected.
- Drop response code (**drop-response-code**) - Response code for drop action.
- Forward to internet (**forward-to-internet**) - Forward the request to the Internet.
- Forward to service-group (**forward-to-service-group**) - Forward the request to the service group.
- Forward to proxy (**forward-to-proxy**) - Forward the request to the HTTP proxy server. This is useful in the case of an upstream proxy server to ensure proper proxy information is not lost during communication.
- Drop (**drop**) - Drop the request.
- Log (**log**)- Log the actions.
- Dual Stack Action (**dual-stack-action**)- Defines the action to forward requests via IPv4 or IPv6 networks. This is useful in IPv4 and IPv6 dual stack forwarding in explicit proxy mode.
- Forward HTTP request (**forward-http-connect-to-icap reqmod-icap**) - Forward HTTP CONNECT request to ICAP server based on the configured template.
- No Client Connection Reuse (**no-client-conn-reuse**) – Configure this option only when the HTTP/HTTPS client will not send multiple requests to different destinations over the same TCP connection between the client and the ACOS Device. Most modern web browsers have connection reuse enabled by default. Thus, this setting typically doesn't apply when clients are web browsers.

In this mode, the explicit proxy inspects only the first HTTP request after establishing a TCP connection and applies the forward policy. All subsequent requests on that TCP connection are not inspected and are tunneled directly to the same destination. This mode is designed for higher performance.

- Require web category (**require-web-category**) - Wait for the web category to be resolved before taking a proxy decision.

- SSLi URL filtering (`ssli-url-filtering`) - Destination filtering controls when the policy is used as a Transparent proxy in SSLi setup.
- SSLi URL filtering SAN (`ssli-url-filtering-san`) - Destination filtering controls based on Certificate SAN when the policy is used as Transparent.
- ACOS Event Log (`acos-event-log`) - HTTP requests forwarded or dropped can be logged to the external and/or internal Syslog servers using this command.
- Local Logging (`local-logging`) – Enable local logging.

Apart from the above rules and conditions, you can configure additional forward proxy objects using the SLB forward-proxy (`slb forward-proxy`) and bind it to the 'destination' rules as the match condition.

For detailed information on all the forward-proxy commands, see *Command Line Interface Reference*.

See Also

- [Configuring Forward Policy in PBSLB Template](#)
- [Configuring and Binding Forward Proxy Objects](#)

Configuring Forward Policy in PBSLB Template

This section describes the CLI examples to configure the forward policy actions in the PBSLB template.

CLI Configuration - Example 1

1. Configure the source and destination rules with only one match condition.

The **HTTP-policy** is configured, which takes the appropriate traffic to its destination through the **match-any** source and destination action. The source network address translation (SNAT) is an additional action.

```

ACOS(config)# slb template policy HTTP-POLICY
ACOS(config-policy)# forward-policy
ACOS(config-policy-forward-policy)# action A1
ACOS(config-policy-forward-policy-action)# forward-to-internet HTTP
snat NAT
ACOS(config-policy-forward-policy-action)# log
ACOS(config-policy-forward-policy-action)# exit
ACOS(config-policy-forward-policy)# source ANY
ACOS(config-policy-forward-policy-source)# match-any
ACOS(config-policy-forward-policy-source)# destination any action A1
ACOS(config-policy-forward-policy-source)# exit

```

2. Bind the SLB template policy to the HTTP virtual port.

```

ACOS(config)# slb virtual-server VS 10.10.10.100
ACOS(config-slb vserver)# port 80 http
ACOS(config-slb vserver-vport)# template policy HTTP-POLICY
ACOS(config-slb vserver-vport)# exit

```

CLI Configuration - Example 2

To configure the source and destination rules with multiple conditions, use the following commands:

1. Configure a class list with the matching rule that can be bound to the destination rule. This is an optional configuration.

The class-list **HOST-match** will match the HOST if the request HTTP HOST header ends with the specified hostname, i.e., **example.com**.

```

ACOS(config)# class-list HOST-match ac
ACOS(config-class list)# ends-with example.com

```

2. Enable the advance match feature in the forward-policy module and configure the forward actions.

NOTE: Enabling the advanced match feature will deactivate all the other kinds of destination rules.

The forward-policy **TEST** is enabled with the advance mode, and **forward-to-internet** is the forwarding action to the **HTTP** service group.

```

ACOS(config)# slb template policy TEST
ACOS(config-policy)# forward-policy
ACOS(config-policy-forward-policy)# enable-adv-match
ACOS(config-policy-forward-policy)# action FWD
ACOS(config-policy-forward-policy-action)# forward-to-internet HTTP
snat HTTP
ACOS(config-policy-forward-policy-action)# log

```

3. Configure the advance match conditions and/or bind the class-list to the destination rule. Additionally, configure the forwarding action.

Two destination rules are configured: **adv-match 1000** and **adv-match 900**.

- **adv-match 1000** destination rule matches the **HOST-match** class-list and the **Office_Hours** time-range when the request arrived. When both the rules match, the request is dropped.
- **adv-match 900** destination rule matches the content-length range of the HTTP request. If it is larger than 1 Mbytes, then the request is forwarded.

In this case, the advanced match rules will traverse from 1000 to 900 until the first match.

```

ACOS(config-policy-forward-policy)# source rule-src-dst
ACOS(config-policy-forward-policy-source)# match-any
ACOS(config-policy-forward-policy-source)# destination adv-match 1000
ACOS(config-policy-forward-policy-source-...)# match-host HOST-match
ACOS(config-policy-forward-policy-source-...)# match-time-range Office_
Hours
ACOS(config-policy-forward-policy-source-...)# action DROP
ACOS(config-policy-forward-policy-source-...)# exit
ACOS(config-policy-forward-policy-source)# destination adv-match 900
ACOS(config-policy-forward-policy-source-...)# match-http-content-
length-range 1048576 to 0
ACOS(config-policy-forward-policy-source-...)# action FWD
ACOS(config-policy-forward-policy-source-...)# exit

```

4. Bind the forward-policy to the HTTP virtual port.

```

ACOS(config)# slb virtual-server VS 10.10.10.100
ACOS(config-slb vserver)# port 80 http
ACOS(config-slb vserver-vport)# template policy TEST
ACOS(config-slb vserver-vport)# exit

```

Configuring and Binding Forward Proxy Objects

This section describes the forward proxy objects that can be configured and bound to the forward-policy template as additional actions.

For creating the forward proxy objects, use the `slb forward-proxy` command in the global configuration and configure the following options:

- **Notify Page (`notify-page`)** - Send the client a notification webpage with a response message. When an HTTP request hits a rule with this additional action, ACOS will notify the client before forwarding the request as the primary action of the rule.
- **Time-Range (`time-range`)** - Defines the time range. When the HTTP request matches a rule with this additional action, ACOS will drop or accept the request.

For detailed information on all the commands, see *Command Line Interface Reference Guide*.

CLI Configuration

1. Configure the notification page and response message.

The **HIT** notify page will display **The match is HIT** response message to the client when an HTTP request a match rule. ACOS will notify the client before forwarding the request as the primary action of the rule.

```
ACOS(config)# slb forward-proxy notify-page HIT
ACOS(config-notify-page:HIT)# message "The match is HIT"
```

2. Setup the time range when the match rules should trigger for the client request.

The **Office_Hours** time-range condition is configured, and the match rule will trigger from **09:00 to 18:00** every **Monday through Friday**.

```
ACOS(config)# slb forward-proxy time-range Office_Hours
ACOS(config-time-range:OfficeHours)# daily 09:00 to 18:00
ACOS(config-time-range:OfficeHours)# weekday-mo
ACOS(config-time-range:OfficeHours)# weekday-tu
ACOS(config-time-range:OfficeHours)# weekday-we
ACOS(config-time-range:OfficeHours)# weekday-th
ACOS(config-time-range:OfficeHours)# weekday-fr
```

3. Enable the advance match mode and bind the forward proxy object as the match

condition.

The **adv-match 950** destination rule matches the **Office_Hours** time-range condition. If it matches, the action is to **FWD** the request and notify the client with the **The match is HIT** response message.

```
ACOS(config)# slb template policy TEST
ACOS(config-policy)# forward-policy
ACOS(config-policy-forward-policy)# enable-adv-match
ACOS(config-policy-forward-policy)# source rule-src-dst
ACOS(config-policy-forward-policy-source)# match-any
ACOS(config-policy-forward-policy-source)# destination adv-match 950
ACOS(config-policy-forward-policy-source-...)# match-time-range Office_
Hours
ACOS(config-policy-forward-policy-source-...)# notify-page
ACOS(config-policy-forward-policy-source-...)# action FWD
```

4. Bind the template policy in the virtual port.

```
ACOS(config)# slb virtual-server VS 10.10.10.100
ACOS(config-slb vserver)# port 80 http
ACOS(config-slb vserver-vport)# template policy TEST
ACOS(config-slb vserver-vport)# exit
```

Viewing Forward Policy Statistics

You can view the all the forward-policy statistics of PBSLB template using **show slb template policy forward-policy-stats** command.

Show Command Example

```
ACOS# show slb template policy forward-policy-stats

slb template policy name: Explicit_Proxy
Source NAT failure:                0
Unresolved DNS requests:          0
Outstanding DNS requests:         0
Hits:                             1
Requests forward to Internet:     0
Requests forward to Service Group: 0
Requests forward to Proxy:        0
Requests dropped:                 1
Source Match not found:           0
Expected Client HELLO requests not found:0
```

Binding Class-List to PBSLB Template

The PBSLB template allows binding system-wide (global) class-list for IP limiting rules feature. This template can then be bound to service group or virtual port. Additionally, you can configure GLID or LID in the class-list.

If a LID or GLID in a class list contains request limiting or request-rate limiting settings, it can only be applied if the following conditions are true:

- The LID or GLID is used within a policy template.
- The policy template is bound to a virtual port.

CLI Configuration

1. Configure the GLID.

```
ACOS(config)# glid 10
ACOS(config-glid:10)# conn-rate-limit 100 per 100
ACOS(config-glid:10)# over-limit-action lockout 1000
ACOS(config-glid:10)# exit
```

2. Configure global class-list with IPv4 host or subnet address of the client in

standard CIDR notation.

```
ACOS(config)# class-list rrl-ipv4
ACOS(config-class list)# 0.0.0.0/0 glid 10
ACOS(config-class list)# exit
```

3. Configure the PBSLB entry age.

```
ACOS(config)# slb common
ACOS(config-common)# pbslb-entry-age 2
ACOS(config-common)# exit
```

4. Configure the PBSLB template.

```
ACOS(config)# slb template policy policy1
ACOS(config-policy)# class-list rrl-ipv4
```

5. Configure the virtual server and bind the 'policy1' template.

```
ACOS(config)# slb virtual-server vs 1.1.1.120
ACOS(config-slb vserver)# port 80 http
ACOS(config-slb vserver-vport)# template policy policy1
ACOS(config-slb vserver-vport)# exit
ACOS(config-slb vserver)# exit
```

Show Command

To view the PBSLB configuration and statistics for service-groups, use **show pbslb client** command.

```
ACOS#show pbslb client
Client class list statistics:
F = Flag (C-Connection, R-Request, S-Response code, D-direct action),
Over-L = Over-limit, Over-RL = Over rate limit, LkupT = Lockup time left,
Lkup# = Lockup number
when F=S; Rcode = response code range, Rate = number of RSP code in period
when F=D; Please use the description with ()
Source Destination Rcode F Current Rate Over-L Over-RL LkupT Lkup# Age
(lid num) (D) drop (Reset) (Ser-sel) (Ser-sel-fail) (Age)
-----+-----+-----+-----+-----+-----+-----+-----+-----+
1.1.1.91 1.1.1.120:80 C 0 0 0 0 0 0 0 5
Total: 1
```

Binding PBSLB Templates to Service Groups

You can configure ACOS to allow some types of client requests to be directed to one service group, using restrictive traffic control policies, while sending all other requests to a separate service group.

In this hypothetical use case scenario, the enhancement could be used to throttle client requests destined for a specific URL while allowing the other requests to flow through the ACOS device unimpeded.

Consider the following high-level configuration:

- Create two overlapping service groups (SG1 and SG2) containing the same real servers.
 - SG1 could have a policy that limits the number of user requests not more than 100 requests per second.
 - SG2 could have no rate limiting policy.
- Create a policy template that uses URL switching. This template could direct requests starting with “/auth” (authentication requests) to the first service group (SG1), while all other requests would be sent to the default service group (SG2).
- Bind the policy template to the VIP.

The outcome of these configurations is that it would effectively throttle just the client requests starting with “/auth”, but all other traffic would be able to reach the default service group, which would not impose any rate limits on that traffic.

This section described GUI and CLI configurations for binding policy templates to service groups.

GUI Configuration

Create a class list and new policy template:

1. Navigate to **ADC >> Templates >> L7 Protocols** tab.
2. Click **+ Create**, then select **Policy** from the pop-up menu.
3. Enter a name for the policy template.
4. Expand the **Class List** section.

- a. Select a previously created class list from the **Class List** field.
- b. Enter the desired rate limiting parameters for this class list, and then click **Add**.

When binding a policy template at the service-group level, only class lists are supported; black/white lists are not supported.

5. When finished adding class lists, click **OK**.

Bind the policy template to the desired service group.

1. Navigate to **ADC >> SLB >> Service Groups** tab.
2. Click **Edit** in the Action column for the service group you want to modify.
3. On the Update Service Group page, expand the **Advanced** Fields section.
4. In the **Template Policy** field, select the policy to bind to this service group from the drop-down list.
5. Click **Update**.

CLI Configuration

The following commands configure a class list called “test1” with multiple LIDs defined:

```
ACOS(config)# class-list test1
ACOS(config-class list)# 192.168.0.200/32 lid 1
ACOS(config-class list)# 192.168.0.100/32 lid 2
ACOS(config-class list)# 192.168.100.0/24 lid 5
ACOS(config-class list)# 0.0.0.0/0 lid 6
ACOS(config-class list)# exit
```

These commands configure a policy template containing a class list “test1” and performs request and connection limiting:

```
ACOS(config)# slb template policy req-limit
ACOS(config-policy)# class-list test1
ACOS(config-policy-class-list:test1)# lid 1
ACOS(config-policy-class-list:test1-...)# conn-limit 2
ACOS(config-policy-class-list:test1-...)# request-limit 3
ACOS(config-policy-class-list:test1-...)# over-limit-action log
ACOS(config-policy-class-list:test1-...)# exit
```

```

ACOS(config-policy-class-list:test1)# lid 2
ACOS(config-policy-class-list:test1-...)# conn-limit 5
ACOS(config-policy-class-list:test1-...)# exit
ACOS(config-policy-class-list:test1)# lid 5
ACOS(config-policy-class-list:test1-...)# conn-limit 8
ACOS(config-policy-class-list:test1-...)# exit
ACOS(config-policy-class-list:test1)# lid 6
ACOS(config-policy-class-list:test1-...)# conn-limit 1
ACOS(config-policy-class-list:test1-...)# exit
ACOS(config-policy-class-list:test1)# exit
ACOS(config-policy)# exit

```

To bind a policy template to a service group, use the **template policy** CLI command at the service group level. The following commands bind the req-limit policy template to the service group called “sg801”:

```

ACOS(config)# slb service-group sg801 tcp
ACOS(config-slb svc group)# template policy req-limit
ACOS(config-slb svc group)# member rs801 80
ACOS(config-slb svc group-member:80)# exit
ACOS(config-slb svc group)# member rs801_1 80
ACOS(config-slb svc group-member:80)# exit
ACOS(config-slb svc group)# member rs801_2 80
ACOS(config-slb svc group-member:80)# end

```

Binding Policy Template to Virtual Ports

PBSLB configuration on a virtual port can be set either using a template or by configuring the individual settings on the port. Individual PBSLB settings *and* a PBSLB template can not be configured on the same virtual port.

ACOS allows you to bind only one PBSLB template to a virtual port. However, you can bind the same PBSLB template to multiple ports.

ACOS device also allows policy templates to be applied at the virtual-server level. However, PBSLB does not take effect if you apply the policy template at the virtual-server level. Only class lists are supported at the virtual-server level. To use PBSLB, apply the policy template globally or on individual virtual ports.

Consider the following high-level configuration example:

- Create TCP and HTTP virtual servers (VStcp and VShttp) containing the same real servers.
 - VStcp can apply only the connection limits.
 - VShttp can either apply request limits or request-rate limits..
- Create a PBSLB template PB1 that includes class-list with connection limits.
- Create a PBSLB template PB2 that includes request limits and request rate limits.
- Bind the PS1 template to the VStcp and PS2 to VShttp.

This section described GUI and CLI configurations for binding policy templates to service groups.

GUI Configuration

Create a class list and new policy template:

1. Navigate to **Shared Objects >> Class Lists**.
2. Click **Create**, then enter the class list parameters.
3. Click **Create** to save the configuration.
4. Navigate to **ADC >> Templates**.
5. Select the **L7 Protocols** tab.
6. Click **Create**, then select **Policy** from the pop-up menu.
7. Enter a name for the policy template.
8. Expand the **Class List** section.
 - a. Select a previously created class list from the **Class List** field.
 - b. Enter the desired rate limiting parameters for this class list, and then click **Add**.
9. When finished adding class lists, click **OK**.

Bind the policy template to the desired virtual port.

1. Navigate to **ADC >> SLB**.
2. Select the **Virtual Servers** tab.

3. Select a virtual server name to edit an existing server or click **Create** to add a new virtual server.
4. If you are adding a new virtual server, enter the general server settings.
5. In the **Virtual Port** section, click **Create**. The Create Virtual Port window appears.
6. Enter the desired parameters in the **Basic** and **Advanced** fields.
7. In the **Template** field, select the previously configured policy to bind to this virtual port from the drop-down list.
8. Click **Update**.

CLI Configuration

1. Create a system-level class-list.

```
ACOS(config)# class-list example-clist
ACOS(config-class list)# 100.1.0.0/16 lid 1
ACOS(config-class list)# exit
```

2. Apply the system-level class list to the "PB1" template with connection limit.

```
ACOS(config)# slb template policy PB1
ACOS(config-policy)# class-list example-clist
ACOS(config-policy-class-list:example-cl...)# lid 1
ACOS(config-policy-class-list:example-cli...)# conn-limit 5
ACOS(config-policy-class-list:example-cli...)# over-limit-action
forward log
ACOS(config-policy-class-list:example-cli...)# exit
ACOS(config-policy-class-list:example-cl...)# exit
ACOS(config-policy)# exit
```

3. Apply the system-level class-list to the 'PB2' template with request limits and request rate limits.

```
ACOS(config)# slb template policy PB2
ACOS(config-policy)# class-list example-clist
ACOS(config-policy-class-list:example-cl...)# lid 1
ACOS(config-policy-class-list:example-cli...)# request-limit 10
ACOS(config-policy-class-list:example-cli...)# over-limit-action
forward log
ACOS(config-policy-class-list:example-cli...)# exit
```

```
ACOS(config-policy-class-list:example-cl...)# exit
ACOS(config-policy)# exit
```

4. Create Layer 4 TCP virtual server 'VStcp' and bind 'PB1' template to the TCP virtual port.

```
ACOS(config)# slb virtual-server VStcp 1.1.1.100
ACOS(config-slb vserver)# port 80 tcp
ACOS(config-slb vserver-vport)# template policy PB1
ACOS(config-slb vserver-vport)# exit
ACOS(config-slb vserver)# exit
```

5. Create Layer 7 HTTP virtual server 'VShttp' and bind 'PB2' template to the HTTP virtual port.

```
ACOS(config)# slb virtual-server VShttp 1.1.1.120
ACOS(config-slb vserver)# port 80 http
ACOS(config-slb vserver-vport)# template policy PB2
ACOS(config-slb vserver-vport)# exit
ACOS(config-slb vserver)# exit
```

Show Command

To view the PBSLB configuration and statistics for service-groups, use **show pbslb virtual-server** command.

```
ACOS#show pbslb virtual-server
Virtual server class list statistics:
F = Flag (C-Connection, R-Request, S-Response code, D-direct action),
Over-L = Over-limit, Over-RL = Over rate limit, LkupT = Lockup time left,
Lkup# = Lockup number
when F=S; Rcode = response code range, Rate = number of RSP code in period
when F=D; Please use the description with ()
Source  Destination  Rcode  F Current Rate Over-L  Over-RL LkupT Lkup# Age
(lid num)              (D) (drop) (Reset) (Ser-sel) (Ser-sel-fail) (Age)
-----+-----+-----+-----+-----+-----+-----+-----+
1.1.1.91  1.1.1.120:80      C 0           0           0           0           0           0           5
Total: 1
```

Viewing PBSLB Statistics

You can view the all the following configuration information and statistics for PBSLB template using `show pbslb` command.

- System level
- Client by IP address
- Service group
- Virtual server

Show Command Example

```
ACOS#show pbslb
Virtual server class list statistics:
F = Flag (C-Connection, R-Request, S-Response code, D-direct action),
Over-L = Over-limit, Over-RL = Over rate limit, LkupT = Lockup time left,
Lkup# = Lockup number
when F=S; Rcode = response code range, Rate = number of RSP code in period
when F=D; Please use the description with ()
Source  Destination  Rcode  F Current Rate Over-L  Over-RL LkupT Lkup# Age
(lid num)              (D) (drop) (Reset) (Ser-sel) (Ser-sel-fail) (Age)
-----+-----+-----+-----+-----+-----+-----+-----+
1.1.1.91  1.1.1.120:80      C 0          0          0          0          0          0          5
Total: 1

Client class list statistics:
F = Flag (C-Connection, R-Request, S-Response code, D-direct action),
Over-L = Over-limit, Over-RL = Over rate limit, LkupT = Lockup time left,
Lkup# = Lockup number
when F=S; Rcode = response code range, Rate = number of RSP code in period
when F=D; Please use the description with ()
Source  Destination  Rcode  F Current Rate Over-L  Over-RL LkupT Lkup# Age
(lid num)              (D) drop) (Reset) (Ser-sel) (Ser-sel-fail) (Age)
-----+-----+-----+-----+-----+-----+-----+-----+
1.1.1.91  1.1.1.120:80      C 0          0          0          0          0          0          5
Total: 1
```


Scan-All-Members Option (Persistence)

The following topics are covered:

Overview	462
Configuring Scan-All-Members	464

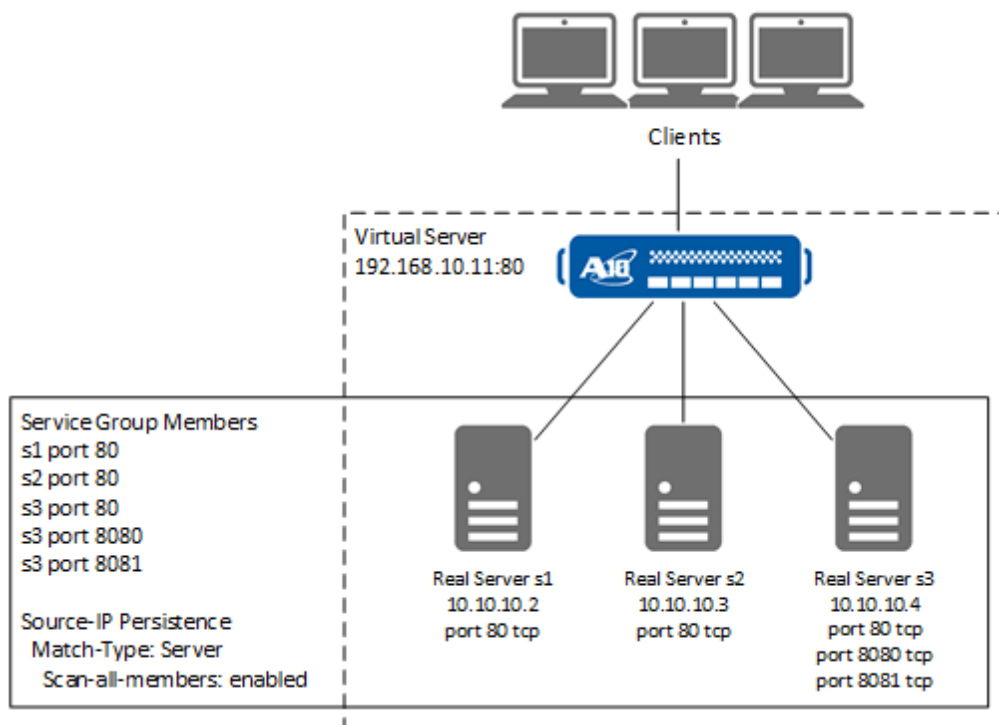
Overview

In cookie, source-IP, and destination-IP persistence templates, the **match-type server** option has a suboption called **scan-all-members**. This option allows a persistent session to continue when the real server port that the session is on becomes unavailable. This section describes the **scan-all-members** option in detail.

The **match-type server** option changes the granularity of persistence, from server+port, to simply server. If the match-type is set to server+port (the default), a persistent session is always sent to the same real port on the same real server. However, if the match-type is set to server, a persistent session is always sent to the same real server, but not necessarily to the same real port.

The **match-type server** option is used in cases where the same service is available on multiple service ports on the server. With this option, if the server port that a client is using for a persistent session goes down, another service port of the same service type on the same server can be used. [Figure 48](#) shows an example.

Figure 48 : Scan-all-members



VIP 192.168.10.11 uses 3 real servers to provide HTTP service. Two of the servers have a single protocol port for HTTP. However, one of the servers has HTTP service on multiple service ports.

For a new session, the SLB load-balancing method enabled on the service group is used to select a server and port for the client (source IP address). Because source-IP persistence is enabled, subsequent requests from the same client are always sent to the same server.

By default, when the match-type is changed to **match-type server**, the ACOS device uses the service group's load-balancing method for the first request to select a service-group member (server+port). For subsequent connections from the same client, the ACOS device uses fast-path processing to select the first service-group member that has the same IP address as the server that was initially selected by the service group's load-balancing method for the first request.

In this example, if the load-balancing method happens to choose port 80 on server s3 for the first request, subsequent connections also are sent to port 80 on s3, since port 80 is in the first member (server+port) for s3 listed in the service-group configuration. Because the match-type is set to **match-type server**, if port 80 goes down, the next request for the persistent session is still sent to s3, but to a different port on s3.

If the load-balancing method selects port 8080 on server s3 for the first request, subsequent requests are sent to port 80 on s3, since port 80 is in the first member (server+port) for s3 listed in the service-group.

However, if the member (port 80 on s3) is set to a lower priority or is administratively disabled, the ACOS device again will use the load-balancing method to select a server and port for the next request. Any of the available service-group members can be selected, even if they are different servers.

It is possible that a different server will be selected for the next request. For example, if an admin needs to perform some maintenance on port 80, and disables that port in order to prevent it from being used for further requests, persistent sessions on the port and server may not remain persistent to the same server.

In a **match-type server** configuration, to ensure that sessions *do* remain persistent on the same server if a member is administratively disabled or is set to a lower priority, use the **scan-all-members** option. In this case, the ACOS device selects the

first available service-group member on the same server as the member that is out of service.

In this example, if s3:80 is disabled or is set to a lower priority, the ACOS device selects the next member on the same server, s3:8080. When s3:80 is available again, it is selected for any new connections. However, connections that are already in existence when s3:80 comes back up continue to go to s3:8080.

Configuring Scan-All-Members

This section described the configuration to scan all members:

GUI Configuration

This procedure configures the source-IP persistence template and service group in [Scan-all-members](#).

Configure the Source-IP Persistence template:

1. Navigate to **ADC >> Templates >> Persistence** tab.
2. Click **Create**, and select **Persist Source IP** from the drop-down list.
3. Enter **persist-source** in the Name field.
4. In the Match Type field, select **Server** from the drop-down list.
5. Select the checkbox in the Scan All Members field.
6. Click **OK**.

Configure the service group:

1. Navigate to **ADC >> Templates >> Service Groups** tab.
2. Click **Create**.
3. Enter **sg-1** in the Name field.
4. In the Member pane, click **Create**.

Add the server s1, port 80 to the service group.

Repeat to add servers s2 (port 80), s3 (port 80), s3 (port 8080), and s3 (port 8081)

5. Click **Create** to create the service group.

Configure the virtual server:

1. Navigate to **ADC >> SLB >> Virtual Servers** tab.
2. Click **Create**.
3. Enter **vip1** in the Name field.
4. Enter **192.168.10.11** in the IP Address field.
5. In the Virtual Port field, click **Create**.
 - a. In the Protocol field, select **TCP**.
 - b. In the Port field, enter **80**.
 - c. In the Service Group field, select **sg-1** from the drop-down list.
 - d. Expand the Templates pane further down on the screen.
 - e. In the Template Persist Source IP field, select **persist-source** from the drop-down list.
 - f. Click **Create**.
6. Click **Create**.

CLI Configuration

These commands configure the source-IP persistence template and service group in [Scan-all-members](#).

The following commands configure the source-IP persistence template:

```
ACOS(config)# slb template persist source-ip persist-source
ACOS(config-source ip persistence template)# match-type server scan-all-
members
ACOS(config-source ip persistence template)# exit
```

The following commands configure the service group:

```
ACOS(config)# slb service-group sg-1 tcp
ACOS(config-slb svc group)# member s1 80
ACOS(config-slb svc group-member:80)# exit
```

```
ACOS(config-slb svc group)# member s2 80  
ACOS(config-slb svc group-member:80)# exit  
ACOS(config-slb svc group)# member s3 80  
ACOS(config-slb svc group-member:80)# exit  
ACOS(config-slb svc group)# member s3 8080  
ACOS(config-slb svc group-member:8080)# exit  
ACOS(config-slb svc group)# member s3 8081  
ACOS(config-slb svc group-member:8081)# exit  
ACOS(config-slb svc group)# exit
```

The following commands configure the virtual server:

```
ACOS(config)# slb virtual-server vip1 192.168.10.11  
ACOS(config-slb vserver)# port 80 tcp  
ACOS(config-slb vserver-vport)# service-group sg-1  
ACOS(config-slb vserver-vport)# template persist source-ip persist-source  
ACOS(config-slb vserver-vport)# exit  
ACOS(config-slb vserver)# exit
```

Quality of Service Marking for TCP Traffic

The following topics are covered:

Overview	468
Configuring Quality of Service for TCP	469

Overview

ACOS provides an option to mark QoS for SLB traffic. The option marks the DSCP (Layer 3) and 802.1p priority (Layer 2) values in client-server SLB traffic. When this feature is configured, ACOS marks traffic in both directions, ACOS-to-client traffic and ACOS-to-server traffic.

The QoS option is disabled by default. You can configure the option in TCP, TCP-proxy, and UDP templates.

When you configure the QoS option, you set a value in the range 1-63. Based on the value you specify, ACOS marks the traffic as follows:

- Layer 3 marking – ACOS sets DiffServ Control Point (DSCP) value in the IP header to a specified value.
- Layer 2 marking – ACOS sets 802.1p value in the MAC header to a specified value divided by 9:
- $dscp-value / 9 = 802.1p-value$

[Table 10](#) lists the 802.1p values ACOS uses based on the configured DSCP value.

Table 10 : SLB QoS Marking Based on DSCP Value

Configured DSCP Value	Marked 802.1p Value
1-7	0
8-15	1
16-23	2
24-31	3
32-39	4
40-47	5
48-55	6
56-63	7

To configure the QoS option, use either of the following methods.

Configuring Quality of Service for TCP

GUI Configuration

1. Navigate to **ADC >> Templates >> L4 Protocols** tab.
2. On the configuration page for the TCP, TCP-proxy, or UDP template, enter the value in the QoS field.
3. Click **Create**.

CLI Configuration

At the configuration level for the TCP, TCP-proxy, or UDP template, use the `qos` command. The following example shows an example in a TCP template:

```
ACOS(config)# slb template tcp tcp-template
ACOS(config-l4 tcp)# qos 55
ACOS(config-l4 tcp)# exit
```

ADC Multi-PU Configuration

ADC implementation is supported on high-performance Multiple Processing Unit (multi-PU) platforms such as Thunder 7650/7655S. These platforms are built for large-scale traffic distribution, load balancing, and scalability. The multi-PU architecture has two active PUs; PU1 (Primary) and PU2 (Blade). The traffic for the same object is distributed to the PUs running the specific object. Both PUs could process the traffic, store the data, files, and compute statistics respectively. This information can be accessed by logging in to the individual PU.

ADC in the Multi-PU platform enables distributing Layer 7 traffic across all PUs, such as:

- HTTP, HTTPS, SSL, DNS-UDP, and DNS-TCP
- HTTP/2 and HTTP 1.1 protocols via IPv4 and IPv6 networks
- All the TLS versions and TLS offloading with backend HTTP/1.1 or HTTP/2

The following topics are covered:

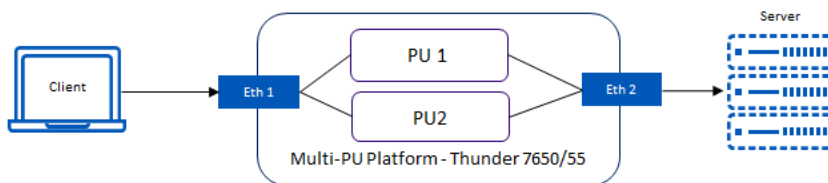
Traffic Flow	470
Key Considerations	472
CLI Configuration	472
Consolidated Configuration Example	477
Limitation	481

For more information on the other product-specific multi-PU configuration such as AAM, GSLB, SSLi, NGWAF, and so on, refer to the respective Configuration Guide.

Traffic Flow

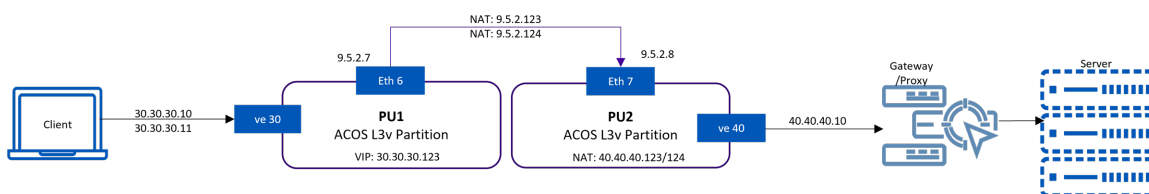
This section provides an example of the multi-PU platform traffic flow using IP address. Here, IP address is the primary way of sending traffic between the PU1 and the PU2.

Figure 49 : Multi-PU Architecture Example



[Figure 49](#) is an example of multi-PU architecture. PU1 and PU2 share all the interfaces. The traffic distribution for this architecture is implemented using the client-side and server-side IP addresses (both sides are required). The client-side traffic will be hashed based on client's source IP, and the server-side traffic will be hashed based on the destination IP. This ensures that the traffic from the same object is distributed to the same PU.

Figure 50 : Multi-PU Explicit Proxy Environment Traffic Flow Example



[Figure 50](#) is an example of SLB or SSLi explicit proxy deployment. The deployment is with two L3v partitions. However, one shared partition or one L3v partition deployment is also supported.

- The input traffic (ve30, eth7) is hashed based on the source IP. Similarly, the output traffic (eth6, ve40) is hashed based on the destination IP. This ensures the same traffic flow is processed on the same PU based on the client's IP. This is achieved by configuring the `traffic-distribution-mode` command under the `interface` or `vlan`. Here, `traffic-distribution-mode sip` is configured for eth7, and `traffic-distribution-mode dip` is configured for eth6.

For more information, see *Network Configuration Guide*.

- The traffic on odd client IP 30.30.30.11 is processed by PU1. ACOS selects 9.5.2.123 for the output traffic source NAT on the inside partition to make sure PU1 also processes the traffic. On the other hand, the traffic on even client IP 30.30.30.10 is processed by PU2. ACOS selects 9.5.2.124 for the output traffic source NAT.

This is achieved by configuring the `odd-even-nat-enable` command under `slb common`. This ensures that the traffic is sent to the same virtual port from two different sources: odd source IP and even source IP. When the traffic from an odd source IP is sent to the same VIP, it is sent to PU1. Similarly, when the traffic from an even source IP is sent to the same VIP, it is sent to PU2.

- For all the source NAT pool-specific configurations, you must configure the `system chassis-port-split` command in the Global configuration mode. This configuration splits the 64k ports into two halves. Moreover, ACOS-initiated traffic from both the PUs allows ACOS to perform DNS queries or AAM requests to the auth-server.
- SLB traffic statistics are aggregated as a single device.

Key Considerations

- Forward and reverse traffic for the same connection needs to be distributed on the same PU.

For example, the `traffic-distribution-mode` options on the client and server interface pair can be SIP/DIP, primary/primary, and secondary/secondary.

- When the `odd-even-nat` configuration is enabled, and the `traffic-distribution-mode` is set to `sip/dip`, each PU acts as an independent unit for the connection limit and connection rate limit configurations. As a result, considering that the traffic is evenly distributed, multi-PU processes twice the amount of traffic than the configured limit. For example, if Connection limit = 100, the total connection limit (PU1+PU2) is actually 200 in the active-active mode.

CLI Configuration

This section describes the CLI configuration guidelines and examples for implementing ADC multi-PU using IP address to ensure seamless traffic load balancing.

Use the following guidelines or workflow to ensure seamless traffic load balancing in both PU1 and PU2:

1. Configure the ADC application type using the `chassis-application-type adc` command.
2. Enable the odd-even IP NAT globally using the `odd-even-nat-enable` command.
3. Configure IP NAT pools using the `ip nat pool` or `ipv6 nat pool` commands to source NAT the client traffic.
4. Configure client-side VLAN (or interface ethernet) with traffic distribution mode SIP using the `traffic-distribution-mode` command.
5. Configure server-side VLAN (or interface ethernet) with traffic distribution mode DIP using the `traffic-distribution-mode` command.
6. Configure the `system chassis-port-split enable` command to split ports for each PU.
7. Create SLB servers, SLB service groups, SLB templates, and SLB virtual servers and ports.
8. Bind the templates and source NAT pool to the virtual ports.
9. Verify the configuration and view the show commands for traffic distribution.

Application Type Configuration

```
ACOS(config)# chassis-application-type adc
```

NOTE:	A reload or reboot is required when the application type is switched from CGN to ADC. Additionally, while switching the box from CGN to ADC, ADC-related configuration should not exist on the box on any partitions (shared/L3V/service).

SLB Global Configuration

```
ACOS(config)# slb common  
ACOS (config-common)# odd-even-nat-enable
```

IP Address and Default Gateway Configuration

```
ACOS(config)# interface management  
ACOS(config-if:management)# ip address 10.65.18.137 255.255.255.0  
ACOS(config-if:management)# ip default-gateway 10.65.18.1
```

```
ACOS(config-if:management) # secondary-ip address 10.65.18.138  
255.255.255.0  
ACOS(config-if:management) # secondary-ip default-gateway 10.65.18.1
```

Interface and VLAN Configuration

```
ACOS(config) # interface ethernet 1  
ACOS(config-if:ethernet:1) # speed-forced-40g  
ACOS(config-if:ethernet:1) # enable  
ACOS(config-if:ethernet:1) # lldp enable rx tx  
ACOS(config-if:ethernet:1) # exit  
ACOS(config) # interface ethernet 2  
ACOS(config-if:ethernet:2) # exit  
ACOS(config) # interface ethernet 3  
ACOS(config-if:ethernet:3) # exit  
....  
ACOS(config) # interface ethernet 5  
ACOS(config-if:ethernet:1) # speed-forced-40g  
ACOS(config-if:ethernet:1) # enable  
ACOS(config-if:ethernet:1) # lldp enable rx tx  
ACOS(config-if:ethernet:1) # exit  
.....  
ACOS(config) # interface ethernet 16  
ACOS(config-if:ethernet:16) # exit  
ACOS(config) # vlan 10  
ACOS(config-vlan:10) # tagged ethernet 1  
ACOS(config-vlan:10) # router-interface ve 10  
ACOS(config-vlan:10) # traffic-distribution-mode sip  
ACOS(config-vlan:10) # exit  
ACOS(config) # vlan 20  
ACOS(config-vlan:20) # tagged ethernet 1  
ACOS(config-vlan:20) # router-interface ve 20  
ACOS(config-vlan:20) # traffic-distribution-mode dip  
ACOS(config-vlan:20) # exit
```

IP Interface on VLAN Configuration

```
ACOS(config) # interface ve 10  
ACOS(config-if:ve:10) # ip address 10.31.8.35 255.255.255.0  
ACOS(config) # interface ve 20
```

```
ACOS(config-if:ve:20)# ip address 10.31.9.35 255.255.255.0
ACOS(config-if:ve:20)# ipv6 address 2602:803:c002::500:142/126
ACOS(config-if:ve:20)# ipv6 enable
ACOS(config-if:ve:20)# exit
```

Common Configuration

```
ACOS(config)# multi-config enable
ACOS(config)# terminal idle-timeout 0
```

Source NAT Pool Configuration

```
ACOS(config)# ip nat pool snat1 10.31.9.48 10.31.9.63 netmask /24
ACOS(config)# ip nat pool snat2 10.31.9.64 10.31.9.79 netmask /24
```

SLB Configuration

```
ACOS(config)# slb server rs1 10.31.9.200
ACOS(config-real server)# port 80 tcp
ACOS(config-real server-node port)# health-check hm_monitor1
ACOS(config-real server-node port)# exit
ACOS(config)# slb server rs2 10.31.9.201
ACOS(config-real server)# port 80 tcp
ACOS(config-real server-node port)# health-check hm_monitor2
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
ACOS(config)# slb service-group sg1 tcp
ACOS(config-slb svc group)# member rs1 80
ACOS(config-slb svc group-member:80)# exit
ACOS(config)# slb service-group sg2 tcp
ACOS(config-slb svc group)# member rs2 80
ACOS(config-slb svc group-member:80)# exit
ACOS(config)# slb template tcp-proxy tcp-default
ACOS(config-policy)# idle-timeout 60
ACOS(config-policy)# exit
ACOS(config)# slb template virtual-port vport_template
ACOS(config-vport)# reset-l7-on-failover
ACOS(config-vport)# exit
ACOS(config)# slb template virtual-server vs_template
ACOS(config-vserver)# conn-limit 5
```

```
ACOS(config-vserver)# conn-rate-limit 10
ACOS(config-vserver)# exit
ACOS(config)# slb template connection-reuse multiplex-default
ACOS(config-conn reuse)# keep-alive-conn 1024
ACOS(config-conn reuse)# exit
ACOS(config)# slb template http http-frp
ACOS(config-http)# insert-client-ip X-Forwarded-For
ACOS(config-http)# request-header-erase X-ClientIP
ACOS(config-http)# keep-client-alive
ACOS(config-http)# exit
ACOS(config)# slb virtual-server vs1 8.43.72.52
ACOS(config-slb vserver)# port 80 http
ACOS(config-slb vserver-vport)# source-nat pool snat1
ACOS(config-slb vserver-vport)# service-group sg1
ACOS(config-slb vserver-vport)# template connection-reuse multiplex-
default
ACOS(config-slb vserver-vport)# template http http-frp
ACOS(config-slb vserver-vport)# template tcp-proxy tcp-default
ACOS(config-slb vserver-vport)# exit
ACOS(config)# slb virtual-server vs2 2602:803:c002:200::43
ACOS(config-slb vserver)# port 80 http
ACOS(config-slb vserver-vport)# source-nat pool snat2
ACOS(config-slb vserver-vport)# service-group sg2
ACOS(config-slb vserver-vport)# template connection-reuse multiplex-
default
ACOS(config-slb vserver-vport)# template http http-frp
ACOS(config-slb vserver-vport)# template tcp-proxy tcp-default
```

Health Check Configuration

```
ACOS(config)# health monitor hm_monitor1
ACOS(config-health:monitor)# interval 10
ACOS(config-health:monitor)# method http expect response-code 200 url GET
/healthcheck
ACOS(config-health:monitor)# exit
ACOS(config)# health monitor hm_monitor2
ACOS(config-health:monitor)# method http port 3000 expect response-code
200 url GET /health
ACOS(config-health:monitor)# exit
```


Consolidated Configuration Example

The consolidated configuration example is given below:

```
ACOS#show run
!Current configuration: 1392 bytes
!Configuration last updated at 23:32:06 PDT Wed Jun 4 2022
!Configuration last saved at 17:45:20 PDT Wed Jun 4 2022
!64-bit Advanced Core OS (ACOS) version 5.2.1-P7-dual-chassis, build 21
(Mar-03-2023,03:26)
!
chassis-application-type adc
!
multi-config enable
!
terminal idle-timeout 0
!
vlan 10
    tagged ethernet 1
    router-interface ve 10
    traffic-distribution-mode sip
!
vlan 20
    tagged ethernet 1
    router-interface ve 20
    traffic-distribution-mode dip
!
interface management
ip address 10.65.18.137 255.255.255.0
ip default-gateway 10.65.18.1
secondary-ip address 10.65.18.138 255.255.255.0
secondary-ip default-gateway 10.65.18.1
enable
```

```
!  
interface ethernet 1  
    speed-forced-40g  
    enable  
    lldp enable rx tx  
!  
interface ethernet 2  
!  
interface ethernet 3  
!  
interface ethernet 4  
!  
interface ethernet 5  
    speed-forced-40g  
    enable  
    lldp enable rx tx  
!  
interface ethernet 6  
    enable  
!  
interface ethernet 7  
    enable  
!  
interface ethernet 8  
!  
interface ethernet 9  
!  
interface ethernet 10  
!  
interface ethernet 11  
!  
interface ethernet 12  
!  
interface ethernet 13  
!  
interface ethernet 14  
!  
interface ethernet 15
```

```
!  
interface ethernet 16  
!  
interface ve 10  
    ip address 10.31.8.35 255.255.255.0  
!  
interface ve 20  
    ip address 10.31.9.35 255.255.255.0  
    ipv6 address 2602:803:c002::500:142/126  
    ipv6 enable  
!  
ip nat pool snat1 10.31.9.48 10.31.9.63 netmask /24  
!  
ip nat pool snat2 10.31.9.64 10.31.9.79 netmask /24  
!  
health monitor hm_monitor1  
    method http expect response-code 200 url GET /healthcheck  
    interval 10  
!  
health monitor hm_monitor2  
    method http port 3000 expect response-code 200 url GET /health  
!  
slb common  
    odd-even-nat-enable  
!  
slb server rs1 10.31.9.200  
    port 80 tcp  
    health-check hm_monitor1  
!  
slb server rs2 10.31.9.201  
    port 80 tcp  
    health-check hm_monitor2  
!  
slb service-group sg1 tcp  
    member rs1 80  
!  
slb service-group sg2 tcp  
    member rs2 80  
!  
slb template tcp-proxy tcp-default
```

```
idle-timeout 60
!
slb template virtual-port vport_template
reset-l7-on-failover
!
slb template virtual-server vs_template
conn-limit 5
conn-rate-limit 10
!
slb template connection-reuse multiplex-default
keep-alive-conn 1024
!
slb template http http-frp
insert-client-ip X-Forwarded-For
request-header-erase X-ClientIP
keep-client-alive
!
slb virtual-server vs1 8.43.72.52
port 80 http
source-nat pool snat1
service-group sg1
template connection-reuse multiplex-default
template http http-frp
template tcp-proxy tcp-default
!
slb virtual-server vs2 2602:803:c002:200::43
port 80 http
source-nat pool snat2
service-group sg2
template connection-reuse multiplex-default
template http http-frp
template tcp-proxy tcp-default
!
```

Limitation

- ADC and CGN cannot co-exist in the box.
- DNS cache synching between PU1 and PU2 is not supported.
- The `odd-even-nat-enable` command is not supported with the `source-nat auto` command.
- Enrollment and renewal of SCEP certificate with IPv6 address is not supported.
- Enrollment and renewal of ACME certificate is not supported.
- With `chassis-application-type adc` configured, if `vrrp-a` is enabled, the VRID-based traffic distribution mechanism is not supported. It is recommended to instead use the `traffic-distribution-mode sip/dip` to distribute traffic based on the source or destination IP address.

Part III: Application Delivery Controller Optimization

This part contains topics pertaining to application offload and optimization.

The following topics are covered:

- [SSL Certificate Management and Options](#)
- [SSL Offload and SSL Proxy](#)
- [Secure FTP Proxy](#)
- [ALG Protocol FWLB Support for FTP and SIP](#)
- [DNS Optimization](#)
- [RAM Caching](#)
- [Transparent Cache Switching](#)
- [Destination Hash-Based TCS](#)
- [Diameter AAA Load Balancing](#)
- [RADIUS Message Load Balancing](#)
- [SNMP-Based Load Balancing](#)
- [Advanced Traffic Replication](#)
- [Outbound Next Hop Load Distributor](#)
- [HTTP Proxy](#)
- [Explicit FTP Proxy](#)
- [Redirection of Traffic to ICAP Servers](#)

SSL Certificate Management and Options

This section describes managing SSL certificates, private keys, and Certificate Revocation Lists (CRLs). An ACOS device can offload SSL processing from servers or, for some types of traffic, can be used as an SSL proxy. (For information about SSL offload and SSL proxy, see [SSL Offload and SSL Proxy](#).)

The following topics are covered:

SSL Certificate Management	485
CA Certificate Versus SSL Certificate	486
Process	486
SSL Templates	491
TLS Server Name Indication (SNI) Support	497
CAs and CSRs Management	503
SSL Certificate Management	485
CA Certificate Versus SSL Certificate	486
Process	486
Certificate Chain	488
Certificate Warning from Client Browser	489
CA-Signed and Self-Signed Certificates	490
SSL Templates	491
Client-SSL Template Configuration and Usage Guidelines	491
Server-SSL Template Configuration and Usage Guidelines	495
Cipher Template Configuration and Usage Guidelines	497
TLS Server Name Indication (SNI) Support	497
Default Certificate and Key	498
SSL Server Certificate Authorization Validation	498
SNI Extension Support	499
Partition Support	500
Configuring TLS Server Name Indication	500

TLS SNI Support on vThunder	501
CAs and CSRs Management	503
Importing a Certificate and Key	504
Generating a Certificate Signing Request (CSR)	506
Generating a Self-Signed Certificate and Key	507
Generating an SSL Certificate – Private Key File with a CSR	509
Installing Certificates	513
Implementing Certificates to SSL Templates	516
Binding Server-SSL Templates to Individual Real Ports	517
Creating Multiple CA Certificate in Server-SSL Templates	519
Configuring Email Notification for SSL Certificate Expiration	521
Converting Certificates and CRLs to PEM Format	522
Importing a Certificate Revocation List (CRL)	524
Exporting Certificates, Keys, and CRLs	524
Importing a CA Cert and Private Key	526
Configuring Forward Proxy Alternate Signing Cert	526
Deleting Certificate Files	527
SSL Certificate Notification via System Log Warnings	528
Configuring Simple Certificate Enrollment Protocol (SCEP) Certificates	528
Configuring Automatic Certificate Management Environment (ACME) Certificates	534

SSL Certificate Management

Some types of client-server traffic need to be encrypted for security. For example, traffic for online shopping must be encrypted to secure sensitive account information from being stolen.

Commonly, clients and servers use SSL or TLS to secure traffic. For example, a client that is using a shopping application on a server will encrypt data before sending it to the server. The server will decrypt the client's data, then send an encrypted reply to the client. The client will decrypt the server reply, and so on. SSL is an older version of TLS.

ACOS device supports the following SSL and TLS versions:

- SSL v3.0
- SSL v3.0
- TLS v1.1
- TLS v1.2
- TLS v1.3

By default, the following RFC standards are supported in ACOS:

- [RFC 3268](#) - AES Cipher suites for TLS. For simplicity, elsewhere this document and other ACOS user documents use the term “SSL” to mean both SSL and TLS.
- [RFC 5746](#) - Renegotiation Indication Extension along with renegotiation_info TLS extension. It allows ACOS to securely renegotiate TLS connections with clients, using existing secure connections. SSL renegotiation is supported on the Software SSL v1.3 module and all the Hardware SSL modules. However, it is not supported on Software SSL TLS v1.2 module.
- [RFC 1421](#) - Privacy Enhanced Mail (PEM) format for certificate files and CRLs. It allows ACOS to process PEM format and RSA encryption.
- [RFC 7627](#) - Extended Master Secret (EMS) (and a TLS extension). It allows ACOS to define a TLS extension that contextually binds the master secret to a log of the full handshake that computes and prevents man-in-the-middle (MITM) attacks.

NOTE: When an SSL certificate expires or is near expiration, the ACOS device will automatically send a system log warning, rather than a system log notice. For information on enabling SNMP traps for SSL certificate events, refer to the *System Configuration and Administration Guide*.

CA Certificate Versus SSL Certificate

Although both terms, *CA certificate* and *SSL certificate*, refer to a certificates used in the SSL protocol, ACOS reserves the term *SSL certificate* for self-signed certificates that are used to create proxied certificates for SSL handshaking with clients in the SSLi, SSL Proxy or SSL offload applications. SSL certificates require a private key to be proxied

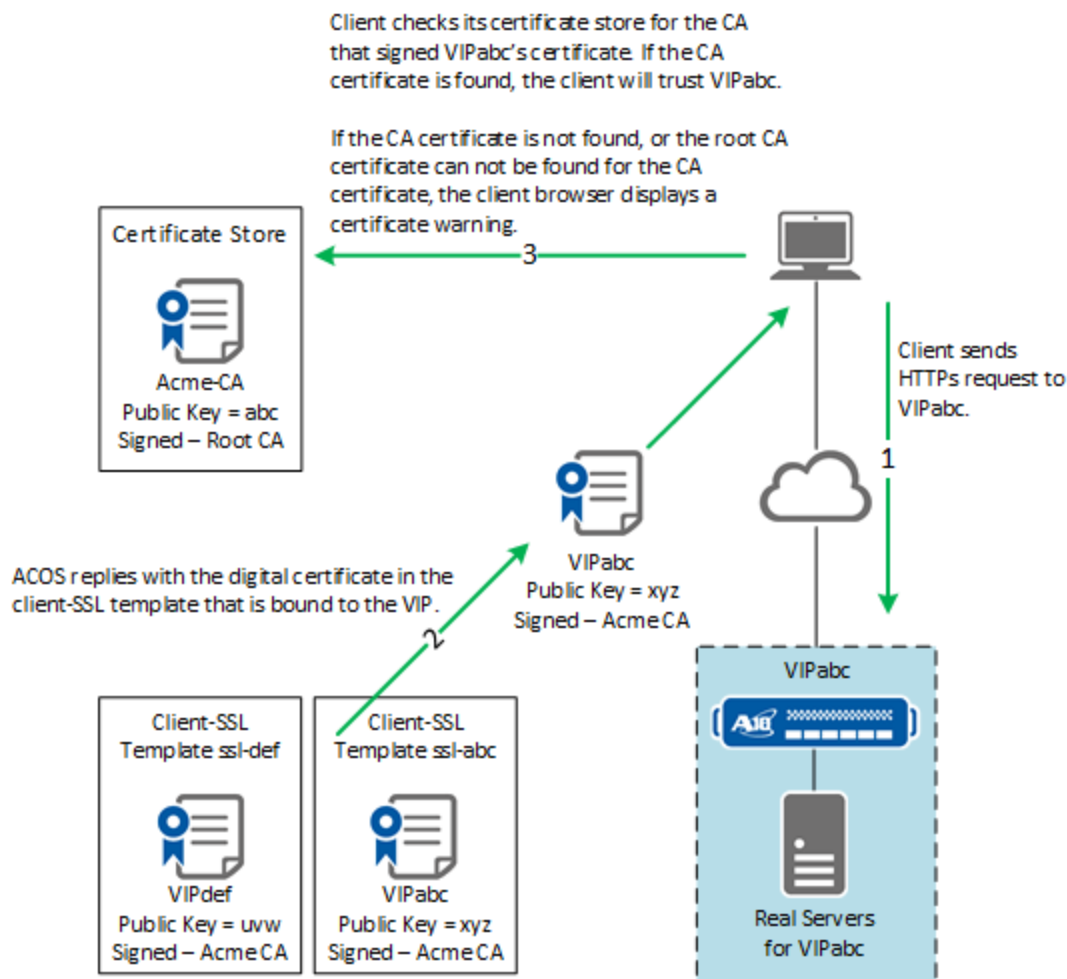
CA certificates are issued by publicly recognized certificate authorities. These certificates are used for other purposes.

Process

SSL works using certificates and keys. Typically, a client will begin a secure session by sending an HTTPS request to a VIP. The request begins an SSL handshake. The ACOS device will respond with a digital certificate, to provide verification of the content server's identity. From the client's perspective, this certificate comes from the server. Once the SSL handshake is complete, the client begins an encrypted client-server session with the ACOS device.

[Figure 51](#) shows a simplified example of an SSL handshake. In this example, the ACOS device is acting as an SSL proxy for backend servers.

Figure 51 : Typical SSL Handshake (simplified)



To begin, the client sends an HTTPS request. The request includes some encryption details such as the cipher suites supported by the client.

The ACOS device, on behalf of the server, checks for a client-SSL template bound to the VIP. If a client-SSL template is bound to the VIP, the ACOS device sends all the digital certificates contained in the template to the client.

The client browser checks its certificate store (sometimes called the certificate list) for a copy of the server certificate. If the client does not have a copy of the server certificate, the client will check for a certificate from the Certificate Authority (CA) that signed the server certificate.

Certificate Chain

Ultimately, a certificate must be validated by a root CA. Certificates from root CAs are the most trusted. They do not need to be signed by a higher (more trusted) CA.

If the CA that signed the certificate is a root CA, the client browser needs a copy of the root CA's certificate. If the CA that signed the server certificate is not a root CA, the client browser should have another certificate or a certificate chain that includes the CA that signed the CA's certificate.

A certificate chain contains the “chain” of signed certificates that leads from the CA to the signature authority that signed the certificate for the server. Typically, the certificate authority that signs the server certificate also will provide the certificate chain. [Figure 52](#) shows an example of a certificate chain containing three certificates:

Figure 52 : SSL Certificate Chain Example

```
-----BEGIN CERTIFICATE-----
ZS9naWYwITAfMAcGBSsOAwIaBBRLa7kolgYMu9BSOJsprEsHiyEFGDAmFiRodHRw
Oi8vbG9nby52ZXJpc2lnbi5jb20vdnNsb2dvMS5naWYwDQYJKoZIhvcNAQEFBQAD
gYEAheIVEe8vArUOZxKkUIGjaYymzJAh8Ty0uUPrikLpQ0IGezByVdbDUJ+HQLGp
2eruTPZpBNADaEfymstIPixrsuCRhyr3Ymsa2rgzwy9kSXeG83H7E7HxRnpDNZ8
l+uzpU/rk4j3bO/JVxPZMnwzMWriPSYgL1EKYcOSKyReACOSQ=
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
ZS9naWYwITAfMAcGBSsOAwIaBBRLa7kolgYMu9BSOJsprEsHiyEFGDAmFiRodHRw
Oi8vbG9nby52ZXJpc2lnbi5jb20vdnNsb2dvMS5naWYwDQYJKoZIhvcNAQEFBQAD
gYEAheIVEe8vArUOZxKkUIGjaYymzJAh8Ty0uUPrikLpQ0IGezByVdbDUJ+HQLGp
2eruTPZpBNADaEfymstIPixrsuCRhyr3Ymsa2rgzwy9kSXeG83H7E7HxRnpDNZ8
l+uzpU/rk4j3bO/JVxPZMnwzMWriPSYgL1EKYcOSKyReACOSQ=
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
ZS9naWYwITAfMAcGBSsOAwIaBBRLa7kolgYMu9BSOJsprEsHiyEFGDAmFiRodHRw
Oi8vbG9nby52ZXJpc2lnbi5jb20vdnNsb2dvMS5naWYwDQYJKoZIhvcNAQEFBQAD
gYEAheIVEe8vArUOZxKkUIGjaYymzJAh8Ty0uUPrikLpQ0IGezByVdbDUJ+HQLGp
2eruTPZpBNADaEfymstIPixrsuCRhyr3Ymsa2rgzwy9kSXeG83H7E7HxRnpDNZ8
l+uzpU/rk4j3bO/JVxPZMnwzMWriPSYgL1EKYcOSKyReACOSQ=
-----END CERTIFICATE-----
```

The certificate chain file and the server certificate files are text files. Each certificate must begin with the “-----BEGIN CERTIFICATE-----” line and end with the “-----END CERTIFICATE-----” line.

The certificate at the top of the certificate chain file is the server certificate signed by an intermediate CA. The next certificate is the intermediate CA's certificate signed by the root CA. The certificate at the bottom of the certificate chain file is the root CA's certificate.

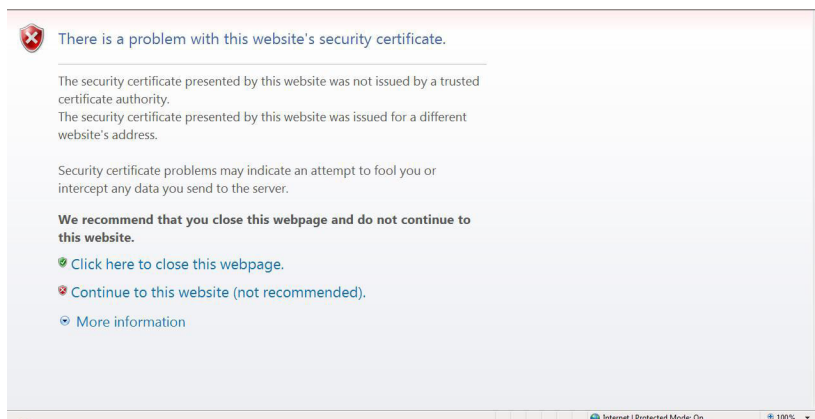
A certificate chain in an SSL template which must begin at the top with the server CA's certificate, followed in order by the intermediary certificate and then the root certificate. If the certificate authority that signs the server certificate does not provide the certificate chain in a single file, you can use a text editor to chain the certificates together in a single file.

Certificate Warning from Client Browser

After the client browser validates the server certificate, the client accepts the certificate and begins an encrypted session with the ACOS device.

If the client can not validate the server certificate or the certificate is out of date, the client's browser may display a certificate warning. [Figure 53](#) shows an example of a certificate warning displayed by Internet Explorer.

Figure 53 : Example of Certificate Warning

**NOTE:**

It is normal for the ACOS device to display a certificate warning when an admin accesses the ACOS management GUI. Certificates used for SLB are not used by the management GUI.

CA-Signed and Self-Signed Certificates

Typically, clients have a certificate store that includes certificates signed by the various root CAs. The certificate store may also have some non-CA certificates that can be validated by a root CA certificate, either directly or through a chain of certificates that end with a root certificate.

Each certificate is digitally “signed” to validate its authenticity. Certificates can be CA-signed or self-signed:

- CA-signed – A CA-signed certificate is a certificate that is created and signed by a recognized Certificate Authority (CA). To obtain a CA-signed certificate, an admin creates a key and a Certificate Signing Request (CSR), and sends the CSR to the CA. The CSR includes the key.

The CA then creates and signs a certificate. The admin installs the certificate on the ACOS device. When a client sends an HTTPS request, the ACOS device sends a copy of the certificate to the client, to verify the identity of the server (ACOS device).

To ensure that clients receive the required chain of certificates, you also can send clients a certificate chain in addition to the server certificate. (See [Certificate Chain](#).)

The example in [Typical SSL Handshake \(simplified\)](#) uses a CA-signed certificate.

- Self-signed – A self-signed certificate is a certificate that is created and signed by the ACOS device. A CA is not used to create or sign the certificate.
- CA-signed certificates are considered to be more secure than self-signed certificates. Likewise, clients are more likely to be able to validate a CA-signed certificate than a self-signed certificate. If you configure the ACOS device to present a self-signed certificate to clients, the client’s browser may display a certificate warning. This can be alarming or confusing to end users. Users can select the option to trust a self-signed certificate, in which case the warning will not re-appear.

SSL Templates

You can install more than one key-certificate pair on the ACOS device. The ACOS device selects the certificate(s) to send a client or server based on the SSL template bound to the VIP. You can bind the following types of SSL templates to VIPs:

Client-SSL template – Contains keys and certificates for SSL-encrypted traffic between clients and the ACOS device. A client-SSL template can also contain a certificate chain.

Server-SSL template – Contains CA certificates for SSL-encrypted traffic between servers and ACOS device.

NOTE: If you replace a certificate and key in a client-SSL or server-SSL template, you must unbind the template from the virtual ports that use it, then rebind the template to the virtual ports, to place the change into effect.

Client-SSL Template Configuration and Usage Guidelines

Use client-SSL templates for deployments in which traffic between clients and the ACOS device will be SSL-encrypted. Client-SSL templates have the following options.

For the simple deployment example in [Typical SSL Handshake \(simplified\)](#), only the first option (Certificate) needs to be configured. You may also need to configure the Certificate chain option.

A client-SSL template can contain up to 128 certificates or certificate chains.

- **Certificate** – Specifies the server certificate that the VIP will send to a client when configured for SSL proxy, SSL offload, or SSLi operation. The client uses this certificate to validate the server's identity. The certificate can be generated on the ACOS device (self-signed) or can be signed by another entity and imported on the ACOS device.

Multiple certificates can be associated to one client-SSL template. We currently support RSA, ECDSA, and SM2 certificates. Each type can have one certificate associated to one client SSL template.

Key specifies the name of a private key for a server certificate. If the CSR used to request the server certificate is generated on the ACOS device, the private key is automatically generated by the ACOS device, and then the private key is used to create the public key sent to the CA in the CSR. Otherwise, the key must be imported. Use the `show pki cert` command to show the list of certificates and private keys stored on the ACOS device.

- Certificate chain – Specifies a named set of server certificates beginning with a root CA certificate, that contains all the intermediary certificates in the authority chain ending with the authority that signed the server certificate. (See [Certificate Chain](#).)
- CA-Certificate – Specifies a CA certificate that the ACOS device can use to authenticate the identity of a client requesting to connect to the ACOS device. If CA certificates are required, they must be imported on the ACOS device. Multiple CA-certificates can be associated with the client-SSL template.

The ACOS device is configured at the factory with the CA certificate bundle file "default_ca_bundle," including common trusted certificate authorities. Use the `show pki ca-cert` command to view the list of ca-certificates.

- Certificate Revocation List (CRL) – Specifies a list of client certificates that have been revoked by the CAs that signed them. This option is applicable only if the ACOS device will be required to validate the identities of clients.

The CRL should be signed by the same issuer as the CA certificate. Otherwise, the client and ACOS device will not be able to establish a connection.

- SSLv2 bypass – Redirects clients who request SSLv2 sessions to the specified service group.
- Client connection-request response – Specifies the ACOS response to connection requests from clients. This option is applicable only if the ACOS device will be required to validate the identities of clients. The response can be one of the following:
 - ignore (default) – The ACOS device does not request the client to send its certificate.
 - request – The ACOS device requests the client to send its certificate. With this action, the SSL handshake proceeds even if either of the following occurs:

- The client sends a NULL certificate (one with zero length).
- The certificate is invalid, causing client verification to fail.

Use this option if you want to trigger an aFlex policy for further processing.

- **require** – The ACOS device requires the client certificate. This action requests the client to send its certificate. However, the SSL handshake does not proceed (it fails) if the client sends a NULL certificate or the certificate is invalid.
- **Require SNI Cert Matched** – Enables an option for Server Name Indication (SNI) validation to prevent unnecessary information leak to the client. You can configure the `require-sni-cert-matched` command in the client-SSL template, and then bind to the SLB virtual server.

When this feature is enabled:

- ACOS verifies if the SNI matches any domain name listed in the server certificates Common Name (CN) or SAN DNS fields. A maximum of 3 certificates can be configured with different algorithm in the client-SSL.
- If not matched, the ACOS immediately disconnects the client without disclosing any certificate or server-related information.

NOTE:

Following are the limitations:

- Supported SSL Modules: software-tls13 module and QAT module.
- The maximum number of SAN DNS type entries in the certificates configured in the client ssl-template is 16.
- Due to infrastructure limitation, replacing a valid certificate with an invalid certificate of the same name via the import command may cause unexpected problems.
- Wildcard certificates are supported. However, only with the complete left-most label and single wildcard (e.g. DNS=*.a10networks.com).
- Fragment (DNS=ww*.a10networks.com) or part of the label other than the left-most label (e.g. DNS=www.*.a10networks.com) or their combinations is not supported.

- **Session cache size** – Specifies the maximum number of cached sessions for SSL session ID reuse.

- Session cache timeout – Sets the maximum number of seconds a cache entry can remain unused before being removed from the cache. Cache entries age according to the ticket age time. The age time is not reset when a cache entry is used.
- Session ticket lifetime – Sets the lifetime for stateless SSL session ticketing. After a client's SSL ticket expires, they must complete an SSL handshake to set up the next secure session with ACOS.
- Close-notify – Specifies whether the ACOS device sends a close_notify message when an SSL transaction ends, before sending a FIN. This behavior is required by certain types of applications, including PHP cgi.
- SSL False Start – Specifies whether SSL False Start is enabled. SSL False Start is an SSL modification used by the Google Chrome browser for web optimization.

NOTE: The following ciphers are not supported with SSL False Start:

SSL3_RSA_DES_64_CBC_SHA
SSL3_RSA_RC4_40_MD5
TLS1_RSA_EXPORT1024_RC4_56_MD5

If no other ciphers but these are enabled in the client-SSL template, SSL False Start handshakes will fail.

- Cipher – Name of a cipher template containing a set of ciphers to use with clients. By default, the client-SSL template's own set of ciphers is used. (See [Cipher Template Configuration and Usage Guidelines](#).)
- Forward proxy options – Options that are used for SSL Insight.
- Authentication username attribute – Specifies the field to check in SSL certificates from clients, to find the client name.
- Cipher Template – Specifies the cipher suites supported by the ACOS device. When the client sends its connection request, it also sends a list of the cipher suites it can support. The ACOS device selects the strongest cipher suite supported by the client that is also enabled in the template, and uses that cipher suite for traffic with the client. For a list of supported ciphers, refer to the `slb template cipher` command in the *Command Line Interface Reference*.

Server-SSL Template Configuration and Usage Guidelines

A server-SSL template is needed only if traffic between the ACOS device and real servers will be encrypted using SSL. In this case, the ACOS device will be required to validate the identities of the servers.

- **CA-Certificate** – Specifies a CA certificate that the ACOS device can use to authenticate the identity of a server the ACOS device is connecting to. If CA certificates are required, they must be imported onto the ACOS device. Multiple CA-certificate can be associated with the server-SSL template. See [Multiple CA Certificate Support in Server-SSL Templates](#).

The ACOS device is configured at the factory with the CA certificate bundle file "default_ca_bundle," including common trusted certificate authorities. Use the `show pki ca-cert` command to view the list of ca-certificates.

- **Certificate** – Specifies a client certificate that the ACOS device will send to a server when requested for client authentication. In SSL proxy and SSL Insight, when a server requests a client's digital certificate, the ACOS device responds on behalf of the client. Following successful authentication, the server and ACOS device communicates over an SSL-encrypted session.
 - In SSL Proxy, the client and ACOS device communicate over a non-encrypted session. From the server's perspective, the server has an encrypted session with the client.
 - In SSL Insight, the client and ACOS device communicate over an encrypted session. From the client's and the server's perspective, the SSL session is fully encrypted.
- **Key** – Specifies a private key for the client certificate.
- **SSL version** – Highest (most secure) version of SSL/TLS to use. The ACOS device supports the following SSL/TLS versions:
 - SSL v3.0
 - TLS v1.0
 - TLS v1.1
 - TLS v1.2 (the default)

- Close notification – Specifies whether the ACOS device sends a close_notify message when an SSL transaction ends, before sending a FIN. This behavior is required by certain types of applications, including PHP cgi.

The close notification option may not work if connection reuse is also configured on the same virtual port. In this case, when the server sends a FIN to the ACOS device, the ACOS device will not send a FIN followed by a close notification. Instead, the ACOS device will send a RST.

- Cipher template – Name of a cipher template containing a set of ciphers to use with servers. By default, the server-SSL template's own set of ciphers is used. (See [Cipher Template Configuration and Usage Guidelines](#).)
- Forward proxy – Enables support for capabilities required for SSL Intercept.
- Session cache size – Specifies the maximum number of cached sessions for SSL session ID reuse.
- Session cache timeout – Sets the maximum number of seconds a cache entry can remain unused before being removed from the cache. Cache entries age according to the ticket age time. The age time is not reset when a cache entry is used.
- Session ticket lifetime – Sets the lifetime for stateless SSL session ticketing. After an SSL ticket expires, the SSL handshake must be performed again in order to set up the next secure session with ACOS.
- Cipher list – Specifies the cipher suites supported by the ACOS device. When the server sends its connection request, it also sends a list of the cipher suites it can support. The ACOS device selects the strongest cipher suite supported by the server that is also enabled in the template and uses that cipher suite for traffic with the server. The same cipher suites supported in client-SSL templates are supported in server-SSL templates, for CA certificates. Support for all of them is enabled by default.

NOTE:

For client certificates, the key length for SSL3_RSA_DES_40_CBC_SHA and SSL3_RSA_RC4_40_MD5 must be 512 bits or less. The TLS1_RSA_EXPORT1024_RC4_56_MD5 and TLS1_RSA_EXPORT1024_RC4_56_SHA ciphers are not supported.

Cipher Template Configuration and Usage Guidelines

A cipher template contains a [list of ciphers](#). A client or server who connects to a virtual port that uses the cipher template can use only the ciphers that are listed in the template.

Optionally, you can assign a priority value to each cipher in the template. In this case, the ACOS device tries to use the ciphers based on priority. If the client supports the cipher that has the highest priority, that cipher is used. If the client does not support the highest-priority cipher, the ACOS device attempts to use the cipher that has the second-highest priority, and so on.

Cipher priority can be 1-100. The highest priority (most favored) is 100. By default, each cipher has priority 1. More than one cipher can have the same priority. In this case, the strongest (most secure) cipher is used.

Key Considerations

- An SSL cipher template takes effect only when applied to a client-SSL template or server-SSL template.
- When you apply (bind) a cipher template to a client-SSL or server-SSL template, the settings in the cipher template override any cipher settings in that client-SSL or server-SSL template.
- Priority values are supported only for client-SSL templates. If a cipher template is used by a server-SSL template, the priority values in the cipher template are ignored.
- Some weak ciphers in the cipher template may be rejected if the back-end server is using the HTTP/2 service, see [HTTP/2 RFC](#).

TLS Server Name Indication (SNI) Support

The ACOS device supports the Server Name Indication (SNI) extension for Transport Layer Security (TLS). The SNI extension enables servers that manage content for multiple domains at the same IP address to use a separate server certificate for each domain. One use case for this feature is supporting a web hosting services. The device supports Static and Dynamic SNI extension support.

ACOS also supports configuring SNI-based filtering without an SSLi template for transparent proxy configurations. For more information, see [URL Filtering Based on SNI](#).

To support SNI extensions, the ACOS device allows you to add multiple certificates to a single client-SSL template, and map individual certificates to their domain names.

Default Certificate and Key

The client-SSL template must contain one certificate and private key pair that is not mapped to a domain. The unmapped certificate and key are the default certificate and key for the template. The ACOS device uses the default template for negotiating the SSL session with the client.

If the client includes the SNI extension in its hello message, the ACOS device uses the certificate that is mapped to the domain requested by the client. Otherwise, the ACOS device uses the default certificate.

SSL Server Certificate Authorization Validation

SSL Server CA validation uses the "`use-client-sni`" command configured in the server SSL template. A10 device can use the client SNI (server-name) while communicating with the backend server. The backend server responds with the appropriate server certificate.

This certificate is validated against the CA configured in the server SSL template. The server responds with a certificate having CN (common name) or SAN (subject alternate name) for the server-name configured in the server SSL template.

CLI Configuration

A new `server-name` option is added to the `slb template server-ssl` configuration command. This new CLI on the server SSL template enables the user to define a server-name directly.

```
ACOS(config)#slb template server-ssl test
ACOS(config-server ssl)# server-name server-name www.test.com
```

This user-defined server name takes precedence over the client SNI defined using `use-client-sni` command. Priority is assigned to server name when both `server-name` and `use-client-sni` is present in the `server-ssl` template.

GUI Configuration

1. Navigate to **ADC >> SSL >> Create >> Server SSL Template** page.
2. To consider user-defined server name priority over client defined SNI, select **Server-name**.

SNI Extension Support

This section describes available SNI extension support methods: Static and Dynamic. SNI Extension. When an SNI extension matches multiple entities, the selection is based on the following precedence:

- SNI extension matches static mapping configured with `server-name` command.
- SNI extension matches static mapping configured with `server-name-regex` command.
- SNI extension matches dynamic mapping.

When an SNI extension does not match any of these entities or the client-hello does not contain an SNI extension, the default cert-key pair is used.

Static SNI Extension Support

You can configure up to 1024 certificate-to-domain mappings in a client-SSL template. Each mapping is configured using the `server-name` or the `server-name-regex` command at the configuration level for the client-SSL template.

Dynamic SNI Extension Support

When dynamic SNI extension support is enabled, a certificate-to-domain mapping is created when a cert and key whose file names include the domain name specified by the client “hello” field of an inbound packet. The number of extensions that can be dynamically support on each virtual port is limited only by hardware restrictions.

SNI extensions use the default certificate and key when a “hello” field contains a domain name for which the device does not contain certificate and key with matching file name.

Dynamic SNI Extension support requires that the cert and key have file extensions .cert and .key, respectively; this is achieved by importing the cert and key files. Use of certs and keys that are generated on the device for Dynamic SNI Extensions is not supported.

Dynamic SNI extension support is enabled by using the `server-name-auto-map` command.

Partition Support

This feature is supported in both the shared partition and L3V private partitions.

Configuring TLS Server Name Indication

GUI Configuration

Before creating the certificate-domain mappings, import the server certificates onto the ACOS device.

1. Navigate to **ADC >> Templates >> SSL >> Create >> Client-SSL** templates.
2. Enter the domain name in the Server Name field.
3. Select the certificate from the Server Certificate drop-down list.
4. Select the certificate's private key from the Server Private Key drop-down list.
5. Click **Add**.
6. Repeat for each mapping.

CLI Configuration

- To map a certificate to a domain, use the `server-name` command at the configuration level for the client-SSL template.

- To enable dynamic SNI extension support, use the `server-name-auto-map` command at the configuration level for the client-SSL template.

TLS SNI Support on vThunder

ACOS provides support for the Server Name Indication (SNI) extension to vThunder models. The SNI is an extension to Transport Layer Security (TLS) that allows a single IP address to host multiple domain names, with a separate certificate for each domain.

The client-SSL template bound to the virtual port can contain multiple certificates. When adding a certificate and key to a client-SSL template, you can specify a domain name (“server name”) that the certificate and key belong to. When a client sends an SSL session setup request to a VIP, ACOS sends the server certificate for the requested domain name based on the configuration in the client-SSL template.

In addition to certificates and keys for individual domain names, a client-SSL template also can contain one “default” certificate and key. If the template does not have a certificate for the domain name requested by the client, ACOS sends the default certificate instead.

- The ACOS configuration does not contain any SLB server certificates by default. The “default” certificate and key in a client-SSL template must be imported or generated in ACOS, then added to the template. If you add them to the template without associating them with a domain name, then they become the default certificate and key for the template.
- The cert, key, and server-name commands include an alternate option for specifying multiple certificates of multiple types (RSA, ECDSA, DSA). Multiple certificates are implemented for terminating SSL/TLS connections from clients with the most secure capability or cipher the client can use.
- SSL Intercept, a feature on some hardware models that uses SNI support, is not supported on vThunder devices. This does not provide SSL Intercept support on vThunder models.

Configuring an SSL VIP TLS SNI (CLI Procedure)

The commands in this section configure an SSL VIP that serves the following domains:

- www.example.com
- www.example2.com
- mail.example.com

This configuration allows the ACOS device to set up secure SSL sessions with a client who sends requests to 192.168.2.69:443. ACOS selects a server certificate to send to the client based on the domain name requested by the client.

This example assumes the certificates and keys were already imported into or generated in ACOS.

The `slb template client-ssl cssl` command configures the client-SSL template and places the CLI in template configuration mode where the following commands are available:

- The `cert` and `key` commands add the default certificate and key.
- The `server-name` command adds certificates, keys, and certificate chains for specific domain names.
 - The “cert2” and “cert3” certificates are used for SSL session setup requests to domains www.example2.com and mail.example.com, respectively.
 - The “def_cert” certificate is for requests to any other domain name, such as www.example.com.

When the `server-name` command includes a chain cert, the SNI SSL ctx is configured with the cert and chain cert. When a default chain cert is defined for the template, the default chain cert is used in place of the configured chain.

These commands configure a default and default-alternate certificate to the SSL1 client-ssl template.

```
ACOS(config)# slb template client-ssl SSL1
ACOS(config-client ssl)# certificate CERT1.pem key CERT1.key
ACOS(config-client ssl)# certificate CERT2.pem alternate key CERT2.key
alternate
```

Configuring an SSL VIP TLS SNI (CLI Example)

These commands bind the client-SSL template to the SSL virtual port:

```
ACOS(config)# slb virtual-server example 192.168.2.69
```

```
ACOS(config-slb vserver)# port 443 ssl
ACOS(config-slb vserver-vport)# template client-ssl cssl
ACOS(config-slb vserver-vport)# exit
```

CAs and CSRs Management

Installing SSL resources on the ACOS device enables the device to provide SSL services on behalf of real servers.

The following topics are covered:

Importing a Certificate and Key	504
Generating a Certificate Signing Request (CSR)	506
Generating a Self-Signed Certificate and Key	507
Generating an SSL Certificate – Private Key File with a CSR	509
Installing Certificates	513
Implementing Certificates to SSL Templates	516
Binding Server-SSL Templates to Individual Real Ports	517
Creating Multiple CA Certificate in Server-SSL Templates	519
Configuring Email Notification for SSL Certificate Expiration	521
Converting Certificates and CRLs to PEM Format	522
Importing a Certificate Revocation List (CRL)	524
Exporting Certificates, Keys, and CRLs	524
Importing a CA Cert and Private Key	526
Configuring Forward Proxy Alternate Signing Cert	526
Deleting Certificate Files	527
SSL Certificate Notification via System Log Warnings	528
Configuring Simple Certificate Enrollment Protocol (SCEP) Certificates	528
Configuring Automatic Certificate Management Environment (ACME) Certificates	534

Importing a Certificate and Key

To import certificate and key files, place them on the PC that is running the ACOS GUI or CLI session, or onto a PC or file server that ACOS can reach and fetch the files.

The following topics are covered:

Importing Individual Files	504
SSL Certificate and Key Files Bulk Import	505

Importing Individual Files

To import an SSL certificate CA certificate, certificate chain, or private key follow these instructions.

Importing Certificates (GUI Procedure)

1. Navigate to **ADC >> SSL Management >> SSL Certificates**.
2. Click **Import** to import a certificate or certificate chain.
3. In the **File Name** field, enter a name for the certificate.
4. In the **Import** field, select the item you want to import
5. In the **Import Certificate from** field, select **Local** to import from a local drive on your management PC, **Remote** to import from a remote location, or **Text** to import from the text box that appears
6. In the **SSL or CA Certificate** field, select either **SSL Certificate** or **CA Certificate**.
7. If you are importing a CA-signed certificate for which you used ACOS to generate the CSR, you do not need to import the key. The key is automatically generated by ACOS when you generate the CSR.
8. In the **Certificate Format** field, select the file format of the certificate you are importing. Certificate and private keys in a single file use the PFX format which is automatically chosen.
9. The **Certificate Source** field provides the location and other fields you need to import the selected item.

10. Decide whether to enable or disable the **Overwrite Existing File** option.
11. Click **Import**.

Importing Certificates (CLI Procedure)

- Use the `import cert` command to import a certificate or certificate chain that you will be using with its private key to create proxied certificates for SSL handshaking with clients in the SSLi, SSL Proxy or SSL offload applications. If you import the cert and its key in a single file use the PFX format.

An example of importing a cert for SSLi is found in [Importing a CA Cert and Private Key](#).

- Use the `import ca-cert` command to import a certificate or a certificate chain for certificates for verifying SSL servers and authenticating clients and other purposes. However the CA cert cannot be used for creating proxied signed certificates for handshaking with clients.

NOTE:

If you are importing a CA-signed certificate for which you used ACOS to generate the CSR, you do not need to import the key. The key is automatically generated by ACOS when you generate the CSR.

- Use the `import cert-key` command to import a private key.

SSL Certificate and Key Files Bulk Import

You can import or export SSL files in bulk, as .tgz archives.

Bulk Import and Export of Certificate and Key Files (CLI Procedure)

To import a .tgz archive of SSL certificate files, key files, or CRL files, use the following commands:

- `import cert` – The archive contains only certificate files.
- `import cert-key bulk` – The archive contains both certificate and key files
- `import crl` – The archive contains only CRL files.
- `import key` – The archive contains only Key files.

Generating a Certificate Signing Request (CSR)

The following procedures generates a CSR that you can send to a server, so that the server can send the CSR to a CA to request a new CA-signed certificate or renew an existing one.

This process also creates a public key - private key pair. The public key is sent in the CSR. The private key used to encrypt the CSR.

Generating a CSR (GUI Procedure)

1. Navigate to **ADC >> SSL Management >> SSL Certificates**.
2. Click **+Create**. The **Create SSL Certificates** dialog window appears.
3. In the **Create As** field, select **CSR**.
4. In the **File Name** field, type the name you certificate that will be provided by the CA.
5. In the **Digest** field, select the hashing algorithm used. The default is **sha1**.
6. In the **Cert Type** field, select **RSA** or **ECDSA** depending on which cryptography standard you want.
7. The **Common Name** field is required.

To create a wild card certificate request, use an asterisk for the first part of the common name. For example, to request a wild card certificate for domain example.com and it sub-domains, enter *.example.com as the common name.

8. The **Division, Organization, Locality, State or Province**, and **Email** fields are optional.
9. Select the **Key Size** or accept the default 1024 bytes.
10. Click **OK**.
11. Verify the newly created SSL certificate appears in the **ADC >> SSL Management >> SSL Certificates** page. Check the matching **Name** and **Common Name** fields. The **Type** should be **key**, and the expiration should match the number of days the cert remains valid. See RFC 6125 for help in reading the **Issuer** field.

Generating a CSR (CLI Example)

- Use `pki create csr` command in global configuration mode to generate an RSA

type of certificate signing request (CSR). In this example, the CSR name is CSR1.

```
ACOS(config)# pki create csr CSR1 generate certtype rsa
input key bits(1024,2048,4096) default 1024:
input Common Name, 1~64:CSR1
input Division, 0~31:
input Organization, 0~63:
input Locality, 0~31:
input State or Province, 0~31:
input Country, 2 characters:US
input email address, 0~64:admin@a10networks.com
ACOS(config)#
```

- To create wildcard certificates, use an asterisk as the first part of the common name. For example, to create a wildcard certificate for domain example.com and its sub-domains, enter the following common name: *.example.com.
- To show the CSR created, use `show pki cert csr1 detail` command.

Generating a Self-Signed Certificate and Key

In the following procedure the certificate file also includes the corresponding private key.

See RFC 6125 for help in filling out some of the following fields.

Generating a Self-Signed Certificate and Key (GUI Procedure)

1. Navigate to **ADC >> SSL Management >> SSL Certificates**.
2. Click **+Create**. The **Create SSL Certificates** dialog window appears.
3. In the **Create As** field, select **Certificate**.
4. In the **File Name** field, type the name you certificate that will be generated.
5. Do not enable **CSR Generate**. This checkbox enable the creation of a CSR.
6. In the **Cert Type** field, select **RSA** or **ECDSA** depending on which cryptography standard you want.
7. The **Common Name** field is required.

NOTE: If you need to create a request for a wildcard certificate, use an asterisk as the first part of the common name. For example, to request a wildcard certificate for domain example.com and its sub-domains, enter the following common name: *.example.com.

8. The **Division, Organization, Locality, State or Province**, and **Email** fields are optional.
9. Enter a number for the **Valid Days** (how many days the key will remain valid) and **Key Size**, or accept the defaults 730 days and 1024 bytes.
10. Click **OK**.
11. Verify the newly created SSL certificate appears in the **ADC >> SSL Management >> SSL Certificates** page. Check matching **Name** and **Common Name** fields. The **Type** should be **certificate/key**, and the expiration should match the number of days the cert remains valid. See RFC 6125 for help in reading the **Issuer** field.

Generating a Self-Signed Certificate and Key (CLI Example)

To generate a self-signed certificate, use the following command at the global configuration level of the CLI:

The `pki create certificate` command generates and initializes a self-signed certificate and key. When creating a self-signed certificate it must be pushed out to inside clients (clients on the internal network). If the certificate is not pushed, the internal hosts get an SSL “untrusted root” error whenever they try to connect.

The key length, common name, and number of days the certificate is valid are required. The other information is optional. The default key length is 1024 bits. The default number of days the certificate is valid is 730.

```
ACOS(config)# pki create certificate enterpriseABC-selfsignd certtype
rsa
input key bits(1024,2048,4096) default 1024:
input Common Name, 1~64: enterpriseABC-selfsignd
input Division, 0~31:
input Organization, 0~63:
input Locality, 0~31:
input State or Province, 0~31:US
input Country, 2 characters:US
input email address, 0~64:
```



```
input valid days, 30~3650, default 730:  
ACOS(config)#
```

To create a wildcard certificate, use an asterisk as the first part of the common name. For example, to create a wildcard certificate for domain example.com and its sub-domains, enter the following common name: *.example.com

Generating an SSL Certificate – Private Key File with a CSR

The following procedures generate an SSL self-signed cert with private key and also generate a CSR that you can send to a publicly recognized CA to register your self-signed SSL certificate.

This process also creates a public key - private key pair. The public key is sent in the CSR. The private key is used to encrypt the CSR and also to create the SSL proxied certificate used in the ACOS SSLi, SSL-Offload, and SSL-Proxy applications.

Generating an SSL Certificate – Private Key File with a CSR (GUI Procedure)

1. Navigate to **ADC >> SSL Management >> SSL Certificates**.
2. Click **+Create**. The **Create SSL Certificates** dialog window appears.
3. In the **Create As** field, select **Certificate**.
4. In the **File Name** field, type the name of the certificate that will be generated.
5. Click the **CSR Generate** box to enable the creation of a CSR.
6. In the **Cert Type** field, select **RSA** or **ECDSA** depending on which cryptography standard you want.
7. The **Common Name** field is required.

NOTE:	If you need to create a request for a wildcard certificate, use an asterisk as the first part of the common name. For example, to request a wildcard certificate for domain example.com and its sub-domains, enter the following common name: *.example.com.

8. The **Division, Organization, Locality, State or Province**, and **Email** fields are optional.
9. Enter a number for the **Valid Days** (how many days the key will remain valid) and **Key Size**, or accept the defaults 730 days and 1024 bytes.

10. Click **OK**.
11. Verify the newly created SSL certificate appears in the **ADC >> SSL Management >> SSL Certificates** page. Check the matching **Name** and **Common Name** fields. The **Type** should be **certificate/key**, and the expiration should match the number of days the cert remains valid. See RFC 6125 for help in reading the **Issuer** field. The GUI does not display the CSR separately.

Generating an SSL Certificate – Private Key File with a CSR (CLI Procedure)

- Use the `pki create cert` command in global configuration mode to generate a self-signed SSL certificate and corresponding CSR. In this example, CSR file name is `csr`, CSR renewal file name is `cert-CSR-both`, the file transport protocol is FTP, and the URL specifying where the CSR is sent is 192.168.1.10.

```
ACOS(config)# pki create cert Cert-CSR-both certtype rsa csr-generate
input key bits(1024,2048,4096) default 1024:
input Common Name, 1~64:Cert-CSR-both
input Division, 0~31:
input Organization, 0~63:
input Locality, 0~31:
input State or Province, 0~31:
input Country, 2 characters: US
input email address, 0~64: admin@a10networks.com
```

- In the above example, the CSR is generated without the root CA extensions. The syntax for the command that creates a CSR with root CA extensions follows:

```
ACOS(config)# pki create cert Cert-CSR-both certtype rsa rootca
```

- If you need to create a wildcard certificate, use an asterisk as the first part of the common name. For example, to create a wildcard certificate for domain `example.com` and its sub-domains, enter the following common name: `*.example.com`.
- Use `show pki csr Cert-CSR-both detail` to show the cert created.
- Use `show pki cert Cert-CSR-both detail` to show the CSR created.

```
ACOS(config)# show pki cert Cert-CSR-both detail
Certificate:
  Data:
    Version: 3 (0x2)
```

```

Serial Number: 13866059162969540330 (0xc06e2357db5986ea)
Signature Algorithm: sha1WithRSAEncryption
Issuer: C=AF, CN= Cert-CSR-both
Validity
    Not Before: Jan 31 05:20:36 2017 GMT
    Not After : Jan 31 05:20:36 2019 GMT
Subject: C=AF, CN=Cert-CSR-both
Subject Public Key Info:
    Public Key Algorithm: rsaEncryption
    Public-Key: (1024 bit)
    Modulus:
        00:96:fc:1d:cc:63:ea:c1:a9:c7:1d:dd:c5:9c:72:
        08:61:27:b7:67:1a:27:c7:f7:39:ca:9c:81:ac:f0:
        f8:05:89:1a:66:25:cf:0b:1e:55:cc:cf:8b:89:91:
        58:c5:e9:8c:b8:44:f1:d5:42:94:b1:e9:5a:a6:10:
        05:28:0d:a2:84:a6:73:a8:64:66:e4:72:cc:c8:1b:
        39:c9:4a:9c:a6:b3:67:e1:4a:d8:9d:a3:fa:bd:7c:
        0e:ad:c1:35:6c:6f:54:68:0a:5f:54:67:61:fd:6a:
        e2:55:2f:85:11:76:f3:96:c0:5c:55:11:63:a6:21:
        41:65:6f:da:67:d5:e8:7e:ff
    Exponent: 65537 (0x10001)
Signature Algorithm: sha1WithRSAEncryption
    7d:ac:29:e8:a9:b5:2f:69:43:d2:a1:8b:7c:6d:8e:b5:21:f8:
    30:cc:7a:4f:61:71:23:87:51:2c:da:ce:89:14:29:55:f3:81:
    97:c0:2f:a7:e3:8a:4b:7d:d2:f7:cb:00:14:ce:91:db:1f:3a:
    db:a0:a0:a9:90:b8:a1:b0:7a:16:e3:54:23:94:e2:48:fb:92:
    36:0c:6d:c4:be:fd:79:77:41:6c:3a:19:3f:72:29:c6:95:f1:
    c5:41:d8:a8:ed:18:2e:ca:66:1a:af:39:16:79:10:03:d6:f0:
    95:10:93:1f:13:c8:96:70:c5:3f:97:8b:96:e1:d5:78:8d:b7:
    c7:0c

```

SHA1

Fingerprint=D5:9A:B6:96:66:5D:B9:77:FE:1F:28:B4:BC:A9:3A:43:5D:2D:C7:98

-----BEGIN CERTIFICATE-----

```

MIIBxjCCAS+gAwIBAgIJAMBuI1fbWYbqMA0GCSqGSIb3DQEBBQUAMCUxCzAJBgNV
BAYTAkFGMRyWfAYDVQQDEw1DZXJ0LUNTU1lib3RoMB4XDTE3MDEzMDEzMjAzNloX
DTE5MDEzMDEzMjAzNlowJTELMakGA1UEBhMCQUYxYjAUBG9NVBAMTDUNlc3Q1NS
LWJvdGgwZG8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAJb8Hcxj6sGp3dxZxy

```

```
CGEnt2caJ8f3Ocqcgazw+AWJGmYlzwseVczPi4mRWMXpjLhE8dVC1LHpWqYQBSgN
ooSmc6hkZuRyzMgbOclKnKazZ+FK2J2j+r18Dq3BNWxvVGgKX1RnYf1q41UvhRF2
85bAXFURY6YhQWVv2mfV6H7/AgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAfawp6Km1
L2lD0qGLfG2OtSH4MMx6T2FxI4dRLNrOiRQpVfOB18Avp+OKS33S98sAFM6R2x86
26CgqZC4obB6FuNUI5TiSPuSngxtxL79eXdBbDoZP3IpxpXxxUHYqO0YLspmGq85
FnkQA9bwlRCTHxPIlnDFP5eLluHVeI23xww=
-----END CERTIFICATE-----
```

key size: 1024

ACOS(config)# **show pki csr Cert-CSR-both detail**

Certificate Request:

Data:

Version: 0 (0x0)

Subject: C=AF, CN=Cert-CSR-both

Subject Public Key Info:

Public Key Algorithm: rsaEncryption

Public-Key: (1024 bit)

Modulus:

```
00:96:fc:1d:cc:63:ea:c1:a9:c7:1d:dd:c5:9c:72:
08:61:27:b7:67:1a:27:c7:f7:39:ca:9c:81:ac:f0:
f8:05:89:1a:66:25:cf:0b:1e:55:cc:cf:8b:89:91:
58:c5:e9:8c:b8:44:f1:d5:42:94:b1:e9:5a:a6:10:
05:28:0d:a2:84:a6:73:a8:64:66:e4:72:cc:c8:1b:
39:c9:4a:9c:a6:b3:67:e1:4a:d8:9d:a3:fa:bd:7c:
0e:ad:c1:35:6c:6f:54:68:0a:5f:54:67:61:fd:6a:
e2:55:2f:85:11:76:f3:96:c0:5c:55:11:63:a6:21:
41:65:6f:da:67:d5:e8:7e:ff
```

Exponent: 65537 (0x10001)

Attributes:

a0:00

Signature Algorithm: sha1WithRSAEncryption

```
7f:2e:82:ef:b8:ed:5d:bc:78:4a:8c:25:5e:df:46:69:11:21:
74:7e:1e:fa:29:08:d0:ea:27:1a:25:fa:4b:ae:e2:78:08:2a:
63:ed:c9:0b:8d:0b:f6:d7:1e:07:10:dc:12:2b:ff:b0:0f:4a:
d6:68:a0:e1:ac:80:8b:d7:bb:f2:a3:6e:e2:74:c6:31:6c:44:
cc:45:c3:f8:2c:85:58:cb:a9:dc:28:bb:3b:72:0f:38:95:68:
1d:f4:09:9b:08:0f:f4:49:a5:9d:4d:91:d1:df:82:6c:63:60:
b8:74:d6:13:67:dd:81:c1:a6:af:ee:fa:22:7b:b2:a4:1e:e3:
```

```

b6:3d
-----BEGIN CERTIFICATE REQUEST-----
MIIBZDCBzgIBADA1MQswCQYDVQQGEwJBRjEWMBQGA1UEAxMNQ2VydC1DU1ItYm90
aDCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEA1vwdzGPqwanHHd3FnHIIYSe3
Zxonx/c5ypyBrPD4BYkaZiXPCx5VzM+LiZFYxemMuETx1UKUuselaphAFKA2ihKZz
qGRm5HLMYBs5yUqcprNn4UrYnaP6vXwOrcElbG9UaApfVGdh/WriVS+FEXbzlsBc
VRFjpiFBZW/aZ9Xofv8CAwEAAaAAMA0GCSqGSIB3DQEBBQUAA4GBAH8ugu+47V28
eEqMJV7fRmkRIXR+HvopCNDqJxol+kuu4ngIKmPtyQuNC/bXHgcQ3BIr/7APStZo
oOGsgIvXu/KjbuJ0xjFsRMxFw/gshVjLqdwouzyDziVaB30CZsID/RJpZ1NkdHf
gmXjYLh01hNn3YHBpq/u+iJ7sqQe47Y9
-----END CERTIFICATE REQUEST-----

```

Installing Certificates

To configure an ACOS device to perform SSL processing on behalf of real servers, you must install a certificate on the ACOS device. This certificate is the one that the ACOS device will present to clients during the SSL handshake. You also must configure a client-SSL template, add the key and certificate to the template, and bind the template to the VIP that will be requested by clients.

You can install a CA-signed certificate or a self-signed certificate (described in [Generating a Certificate Signing Request \(CSR\)](#) and [Generating a Self-Signed Certificate and Key](#)).

This section gives an overview of the process for each type of certificate. Detailed procedures are provided later in this section.

Requesting and Installing a CA-Signed Certificate

To request and install a CA-signed certificate, use the following process. For detailed steps, see [CAs and CSRs Management](#) and [Importing a Certificate and Key](#).

1. Create an encryption key.
2. Create a Certificate Signing Request (CSR).

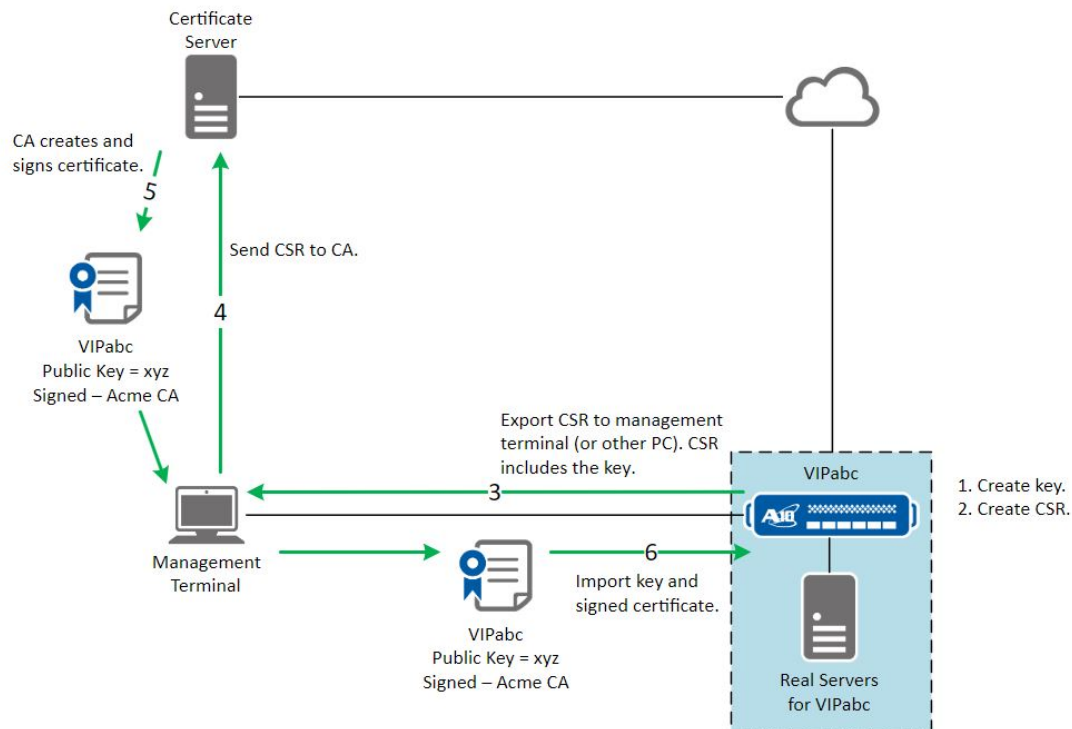
The CSR includes the public portion of the key, as well as information you enter when creating the CSR.

You can create the key and CSR on an ACOS device or a server running openssl or a similar application.

3. Submit the CSR to the CA.
4. If the CSR was created on the ACOS device, do one of the following:
 - Copy and paste the CSR from the ACOS CLI or GUI onto the CSR submission page of the CA server.
 - Export the CSR to another device, such as the PC from which you access the ACOS CLI or GUI. Email the CSR to the CA, or copy-and-paste it onto the CSR submission page of the CA server.
5. If the CSR was created on another device, email the CSR to the CA, or copy-and-paste it onto the CSR submission page of the CA server.
6. After receiving a signed certificate and the CA's public key from the CA, import them to the ACOS device.
7. If the key and certificate are provided by the CA in separate files (PKCS #7 format), import the certificate. The key does not need be imported if the CSR was created on the ACOS device because the key is already on the ACOS device. If the certificate is not in PEM format, specify the certificate format (type) when importing it.
8. If the CSR was not created on the ACOS device, you do need to import the key also.
9. If the key and certificate are provided by the CA in a single file (PKCS #12 format), specify the certificate format (type) when you import it. If the CSR was not created on the ACOS device, you need to import the key also. See [Converting SSL Certificates to PEM Format \(Windows PC Procedure\)](#).
10. If applicable, import the certificate chain onto the ACOS device. The certificate chain must be a single text file, beginning with a root CA's certificate at the top, followed in order by each intermediate signing authority's certificate. (See [Certificate Chain](#).)

[Figure 54](#) shows the most common way to obtain and install a CA-signed certificate onto the ACOS device. You also may need to install a certificate chain file.

Figure 54 : Obtaining and Installing Signed Certificate from CA



NOTE: As an alternative to using a CA, you can use an application such as openssl to create a certificate, then use that certificate as a CA-signed certificate to sign another certificate. However, in this case, a client's browser is still likely to display a certificate warning to the end user.

Installing a Self-Signed Certificate

To install a self-signed certificate instead of a CA-signed certificate:

- Create an encryption key.
- Create the certificate.

See [Generating a Self-Signed Certificate and Key](#).

Implementing Certificates to SSL Templates

After creating or importing certificates and keys on the ACOS device, you must add them to an SSL template, then bind the template to a VIP, in order for them to take effect.

Creating an SSL Template (GUI Procedure)

1. Navigate to **ADC >> Templates >> SSL**.
2. Click **Create**, and:
3. Select **Client SSL** to create a template for SSL traffic between the ACOS device (VIP) and clients.
4. Select **Server SSL** to create a template for SSL traffic between the ACOS device and servers.
5. Enter or select the configuration options; refer to the online help for information about the fields on this GUI page.
6. When finished, click **OK**.

Creating an SSL Template (CLI Example)

Use one of the following commands at the global configuration level of the CLI:

`slb template client-ssl` – creates template for SSL traffic between ACOS device (VIP) and clients.

```
ACOS(config)# slb template client-ssl TMPLT-C
ACOS(config-client ssl)# exit
```

`slb template server-ssl` – creates template for SSL traffic between ACOS device and servers.

```
ACOS(config)# slb template server-ssl TMPLT-S
ACOS(config-server ssl)# exit
```

The command creates the template and changes the CLI to the configuration level for it. Use the commands at the template configuration level to configure template parameters. (For information, see [Generating an SSL Certificate – Private Key File with a CSR](#) or the CLI Reference.)

Binding an SSL Template to a VIP (GUI Procedure)

1. Navigate to **ADC >> SLB > Virtual Servers**.
2. Click Create to create a new virtual server.
3. Enter the VIP name and IP address.
4. In the Port section, click Create. The Virtual Server Port page appears.
5. Click on “Templates” to expand the Templates section.
6. Select the template from the Client-SSL Template or Server-SSL Template drop-down list.

Binding an SSL Template to a VIP (CLI Example)

- Use one of the following commands at the configuration level for the virtual port on the VIP:

template client-ssl – binds client SSL template to the VIP.

```
ACOS(config)# slb virtual-server VIP-1 10.10.1.1
ACOS(config-slbg vserver)# port 80 ssl-proxy
ACOS(config-slbg vserver-vport)# template client-ssl TMPLT-C
ACOS(config-slbg vserver-vport)# exit
ACOS(config-slbg vserver)# exit
```

template server-ssl – binds server SSL template to the VIP.

```
ACOS(config)# slb virtual-server VIP-2 10.10.2.1
ACOS(config-slbg vserver)# port 80 ssl-proxy
ACOS(config-slbg vserver-vport)# template server-ssl TMPLT-S
ACOS(config-slbg vserver-vport)# exit
ACOS(config-slbg vserver)# exit
```

- Use the same command on each port for which SSL will be used.

Binding Server-SSL Templates to Individual Real Ports

For additional flexibility, the ACOS device supports binding of server-SSL templates to individual real ports. This configuration option is useful in cases where the real servers load balanced by a VIP have different SSL settings.

If a server-SSL template is bound to the virtual port instead, all the real servers load balanced by the VIP must use the same SSL settings.

You can bind a server-SSL template to a real port and also to a virtual port that uses that real port. In this case, the server-SSL template bound to the real port is used for traffic sent to that real port. If you remove the server-SSL template from the real port, the template bound to the virtual port is used instead.

Binding Server SSL Templates to Real Ports (GUI Procedure)

On the configuration page for the real server, in the Port section, select the template from the Server-SSL Template drop-down list.

Binding Server SSL Templates to Real Ports (CLI Procedure)

To bind a server-SSL template to a real port, use the `template server-ssl` command at the configuration level for the real port:

Binding Server SSL Templates to Real Ports (CLI Example)

The following commands import a CA-signed certificate and key:

```
ACOS(config)# import ca-cert CACert88.pem tftp:
Address or name of remote host []?192.168.52.254
File name [/]?CACert88.pem
.0 minutes 1 seconds
ACOS(config)# import key CAkey tftp:
Address or name of remote host []?192.168.52.254
File name [/]?CAkey88
.0 minutes 1 seconds
```

The following commands create a server-SSL template and add the certificate and key to the template:

```
ACOS(config)# slb template server-ssl server-ssl1
ACOS(config-server ssl)# ca-cert CACert88.pem key CAkey88
ACOS(config-server ssl)# certificate Cert123.pem key key123 pass-phrase
Pass123
ACOS(config-server ssl)# exit
```

The following commands bind the server-SSL template directly to a port on a real server:

```
ACOS(config)# slb server rs88 10.8.8.8
```

```
ACOS(config-real server)# port 443 tcp
ACOS(config-real server-node port)# template server-ssl server-ssl1
```

NOTE: New certificate commands will be lost if downgrade one box to old image that not support new commands.

Creating Multiple CA Certificate in Server-SSL Templates

If you need to add multiple certificates to a server-SSL template, this section describes how to configure it. A server-SSL template can have multiple CA-signed certificates.

You can add the CA certificates to the server-SSL template in either of the following ways:

- As separate files (one for each certificate)
- As separate files (one for each certificate)

Adding multiple certificates in a single file can simplify configuration. For example, you can export the CA certificates from a web browser into a single file, then import that file onto the ACOS device and add it to a server-SSL template.

Previous releases allow a server-SSL template to have only a single CA-signed certificate.

NOTE: A CA-signed certificate is a certificate signed by a Certificate Authority (CA).

Multiple Certificates in Single File – Preparing the File

You can create the multiple certificate file by exporting the certificates from a browser or you can assemble the file by hand.

To export the certificates from Internet Explorer (IE) version 9:

1. Select **Tools > Internet Options**.
2. Click on the **Content** tab.
3. Click **Certificates**.

4. Click on the **Trusted Root Certification Authorities** tab.
5. Select all the certificates.
6. Click **Export**.
7. Click **Next**.
8. Select PKCS #12 format (PFX), if not already selected.
9. Click **Next**.
10. When prompted for a file password, enter a password to secure the certificate file, and click **Next**.
11. When prompted for a filename:
 - a. Click **Browse** to navigate to the save location for the file.
 - b. Enter the filename and click **Save**.
12. Click **Next**.
13. Click **Finish**.
14. On the ACOS device:
 - a. Import the certificate file as a PFX file.
 - b. Use the GUI or CLI to add the certificate file to a server-SSL certificate.
 - c. Bind the server-SSL certificate to the virtual port.

To create the file manually

1. Copy and paste each certificate to a text file. Make sure to include the "-----BEGIN CERTIFICATE-----" and "-----END CERTIFICATE-----" lines for each certificate. For example:

```
-----BEGIN CERTIFICATE-----  
MIIE0zCCA7ugAwIBAgIQGNr  
RniZ96LtKIVjNzGs7SjANBg  
kqhkiG9w0BAQUFADCB  
-----END CERTIFICATE-----  
-----BEGIN CERTIFICATE-----  
U2lnbiwgSW5jLiAtIEZvciBhd  
XRob3JpemVkIHVzZSBvbmx  
5MUUwQwYDVQQDEzxW
```

```
-----END CERTIFICATE-----
```

2. Save the text file.
3. On the ACOS device:
 - a. Import the certificate file as a PEM file.
 - b. Use the GUI or CLI to add the certificate file to a server-SSL certificate.
 - c. Bind the server-SSL certificate to the virtual port.

Configuring Email Notification for SSL Certificate Expiration

The ACOS device can send email notification when an SSL certificate is about to expire. This feature sends a daily email listing the certificates that are about to expire or that have recently expired.

By default, this feature is not configured. To configure email notification for certificate expiration, use either of the following methods.

Configuring Email Notification for SSL Certificate Expiration (GUI Procedure)

1. Navigate to **ADC >> SSL Management >> Expiration Mail**.
2. In the SSL Expire Email Address, enter the email address (twice; both address must match) where you want the notifications to be sent.
3. Configure the other fields on this screen as desired; refer to the GUI online help for more information about the fields on this page.
4. Click **Update**.

Configuring Email Notification for SSL Certificate Expiration (CLI Procedure and Example)

To configure email notification for certificate expiration, use the `slb ssl-expire-check` command.

The following example enables certificate notifications to be sent to email address "admin1@example.com". Expiration notifications are sent beginning 4 days before expiration and continue for 3 days after expiration.

```
ACOS(config)# slb ssl-expire-check email-address admin1@example.com  
before 4 interval 3
```

Converting Certificates and CRLs to PEM Format

The ACOS device supports Privacy Enhanced Mail (PEM) format for certificate files and CRLs.

If a certificate or CRL you plan to import onto the ACOS device is not in PEM format, it must be converted to PEM format.

You do not need to convert the certificate into PEM format before importing it. You can specify the format when you import the certificate. The ACOS device automatically converts the imported certificate into PEM format. (See [Importing a Certificate and Key](#).)

If you prefer to convert a certificate before importing it, see the following sections.

If you have certificates that are in Windows format, use the procedure in this section to convert them to PEM format. For example, you can use this procedure to export SSL certificates that were created under a Windows IIS environment, for use on servers that are running Apache.

This procedure requires a Windows PC and a Unix/Linux workstation. Perform [Start the Microsoft Management Console \(mmc.exe\)](#), through [Select Action > All Tasks > Export](#), on the Windows PC. Perform [Copy the PFX-format file that was created by the Export wizard to a UNIX machine](#), through [To remove the passphrase from the key, use the following command](#): on the Unix/Linux workstation.

Converting SSL Certificates to PEM Format (Windows PC Procedure)

1. Start the Microsoft Management Console (mmc.exe).
2. Add the Certificates snap-in:
3. Select File Add/Remove Snap-In. The Add/Remove Snap-In dialog appears.
4. Click **Add**. A list of available snap-ins appears.
5. Select Certificates.
6. Click **Add**.
7. A dialog appears with the following choices: My user account, Service account, and Computer account.
8. Select Computer Account and click Next. The Select Computer dialog appears.

9. Select Local Computer and click Finish.
10. Click **Close**.
11. Click **OK**. The Certificates snap-in appears in the Console Root list.
12. Expand the Certificate folders and navigate to the certificate you want to convert.
13. Select **Action > All Tasks > Export**.

The Export wizard guides you with instructions. Make sure to export the private key too. The wizard will ask you to enter a passphrase to use to encrypt the key.

Converting SSL Certificates to PEM Format (Unix / Linux Workstation Procedure)

1. Copy the PFX-format file that was created by the Export wizard to a UNIX machine.
2. Use OpenSSL to convert the PFX file into a PKCS12 format:

```
$ openssl pkcs12 -in filename.pfx -out pfxoutput.txt
```
3. This command creates a PKCS12 output file, which contains a concatenation of the private key and the certificate.
4. Use the vi editor to divide the PKCS12 file into two files, one for the certificate (.crt) and the other for the private key.
5. To remove the passphrase from the key, use the following command:

```
$ openssl rsa -in encrypted.key -out unencrypted.key
```

Although removing the passphrase is optional, A10 Networks, Inc. recommends that you remove the passphrase for production environments where Apache must start unattended.

Converting CRLs from DER to PEM Format (Unix / Linux Workstation Procedure)

If you plan to use a Certificate Revocation List (CRL), the CRL must be in PEM format.

To convert Distinguished Encoding Rules (DER) format to PEM format, use the following command on a Unix/Linux machine where the file is located:

```
openssl crl -in filename.der -inform der -outform pem -out filename.pem
```

Importing a Certificate Revocation List (CRL)

To import a CRL, place it on the PC that is running the GUI or CLI session, or onto a PC or file server that can be locally reached over the network.

Importing a CRL (GUI Procedure)

1. Navigate to **ADC >> SSL Management >> Cert Revocation List**.
2. Click **Import**.
3. Complete the fields on this page to navigate to the location of the CRL.
4. Click **Import**.

Importing a CRL (CLI Procedure)

To import a CRL, use the `import crl` command at the Privileged EXEC or global Config level of the CLI:

Refer to the *Command Line Interface Reference* for detailed information about this command.

Exporting Certificates, Keys, and CRLs

This section describes how to export SSL resources from the ACOS device to other devices.

Due to a limitation in Windows, it is recommended to use names shorter than 255 characters. Windows allows a maximum of 256 characters for both the file name and the directory path. If the combination of directory path and file name is too long, Windows will not recognize the file. This limitation is not present on machines running Linux/Unix.

Exporting a Certificate and Key (GUI Procedure)

1. Navigate to **ADC >> SSL Management >> SSL Certificates**.
2. To export a certificate:
3. Select the certificate. (Click the checkbox next to the certificate name.)
4. Click **Export**.

5. If the browser security settings normally block downloads, you may need to override the setting. For example, in Internet Explorer, hold the Ctrl key while clicking Export.
6. Click **Save**.
7. Navigate to the save location.
8. Click **Save** again.
9. To export a key:
10. Select the key.
11. Click **Export**.
12. Click **Save**.
13. Navigate to the save location.
14. Click **Save** again.

Exporting a Certificate and Key (CLI Procedure)

To export a certificate and its key, use the following commands at the Privileged EXEC or global Config level of the CLI:

```
export cert
```

```
export cert-key
```

Refer to the *Command Line Interface Reference* for detailed information about these commands.

Exporting a CRL (CLI Procedure)

To export a CRL, use the `export crl` command at the Privileged EXEC or global Config level of the CLI:

Exporting a CRL (GUI Procedure)

1. Navigate to **ADC >> SSL Management >> Cert Revocation List**.
2. Select the CRL. (Click the checkbox next to the CRL name.)
3. Click **Export**.
4. If the browser security settings normally block downloads, you may need to override the setting. For example, in IE, hold the Ctrl key while clicking Export.

5. Click **Save**.
6. Navigate to the save location.
7. Click **Save** again.

Importing a CA Cert and Private Key

Import a self-signed CA certificate and the certificate's private key (CLI Example)

The following commands import a self-signed CA certificate trusted by the clients, and the certificate's private key:

```
ACOS-Inside(config)# import cert enterpriseABC-selfsignd scp:  
Address or name of remote host []?192.168.1.111  
User name []?admin  
Password []?*****  
File name [/]?enterpriseABC-selfsignd.pem  
ACOS-Inside(config)# import key enterpriseABC-key scp:  
Address or name of remote host []?192.168.1.111  
User name []?admin  
Password []?*****  
File name [/]?enterpriseABC-key.pem
```

Configuring the client-SSL template to enable SSLi (CLI Example)

The following commands configure the client-SSL template to enable SSLi (forward-proxy). It also specifies the certificate and private key that the inside ACOS Device uses to dynamically create (and cache) forged server certificates as clients request SSL sessions with external servers.

```
ACOS-Inside(config)# slb template client-ssl SSLInsight_ClientSide  
ACOS-Inside(config-client ssl)# forward-proxy-ca-certificate Cert123.pem  
key key123  
ACOS-Inside(config-client ssl)# forward-proxy-enable
```

Configuring Forward Proxy Alternate Signing Cert

In the following example, the inside ACOS Device is configured with a trusted CA list and an alternate signing key. When a client requests connection to an external SSL server, the inside ACOS Device determines whether the certificate of SSL site is

signed by a trusted CA. If it is not in the trusted list, the inside ACOS Device signs the certificate with the alternate signing key. Because the alternate signing key is not trusted, the client will be warned that the site is insecure.

Forward Proxy Alternate Signing Cert (CLI Example)

1. Import the list of trusted list of CAs:

```
ACOS-Inside(config)# import cert ca-cert enterpriseABC-trusted-CAs scp:
...
```

2. Import the list of alternate certificate and signing key:

```
ACOS-Inside(config)# import cert alt-cert scp:
...
ACOS-Inside(config)# import key alt-key scp:
...
```

3. Bind the list of trusted CAs and the alternate signing key to the Client SSL template (which in turn is bound to the SSLi virtual port of the inside ACOS Device.)

```
ACOS-Inside(config)# slb template client-ssl SSLInsight_ClientSide
ACOS-Inside(config-client ssl)# forward-proxy-ca-certificate
Cert123.pem key key123
ACOS-Inside(config-client ssl)# forward-proxy-enable
ACOS-Inside(config-client ssl)# forward-proxy-trusted-ca enterpriseABC-
trusted-CAs
ACOS-Inside(config-client ssl)# forward-proxy-alt-sign cert
```

Deleting Certificate Files

To delete SSL files, use either of the following methods.

SSL File Delete (GUI Procedure)

1. Navigate to one of the following:
 - **ADC >> SSL Management > SSL Certificates**
 - **ADC >> SSL Management > Cert Revocation List**
2. Select the files to delete.

3. Click **Delete**.

SSL File Delete (CLI Procedure)

Using the CLI, you can delete specific SSL files by name.

Use the `pki delete` command at the global configuration level of the CLI to delete SSL files.

SSL Certificate Notification via System Log Warnings

When an SSL certificate expires or is near expiration, the ACOS device will automatically send a system log warning, rather than a system log notice.

For information on enabling SNMP traps for SSL certificate events, refer to the *System Configuration and Administration Guide*.

Configuring Simple Certificate Enrollment Protocol (SCEP)

Certificates

SCEP is a part of the Public key infrastructure (PKI); it simplifies management of security certificates by providing simplified installation and automated renewal of x.509 certificates. You can use SCEP certificates with the same ACOS features that support manually imported certificates. For example, SCEP certificates are supported with SSLi.

NOTE:	This feature is not supported for HSM platforms, including Thunder 5630.

Key Considerations

- SCEP supports shared and L3V partitions. SCEP traffic follows the routing table rule and DNS setting on the shared partition. Hence, the network interfaces, routing rules, and DNS must be configured correctly in the shared partition, as the SCEP certificate is created in the L3V partition.
- To configure a SCEP certificate, you need to specify the certificate name, a password, and the location (URL) of the ES. ACOS handles the rest.

- To use the certificate, add it to an SSL template and bind the template to the virtual port in your application.
- There is no GUI support for configuring this feature.

Enrollment and Renewal Process

After you configure a SCEP certificate for enrollment, ACOS performs the following steps:

1. Generate a private key. In this step, an RSA key with the specified key length is generated for the certificate.
2. Fetch CA certificates. ACOS queries the ES for its certificates. In this step, three certificates are returned: 1 CA certificate and 2 ES certificates, and ES-encryption certificate and an ES-signature certificate.
3. Generate Certificate Signing Request (CSR). The CSR includes the SCEP password you assign to the SCEP certificate, and other parameters needed for the certificate.
4. Fetch the certificate. The CSR is encrypted using the public key of the ES-encryption certificate, and forwarded to the ES.
5. The ES validates the CSR and forwards the request to the CA. The CA then returns the signed certificate. The certificate is signed using the ES-signature certificate.
6. Store the certificate. After successful verification of the response from the CA, ACOS accepts the certificate and stores it. SCEP certificates are stored in DER format. SCEP keys are stored in PEM format.
7. Schedule renewal. ACOS handles automatic renewal of the certificate when its about to expire. ACOS checks the expiration dates of both the enrolled certificate and the issuing CA's certificate. ACOS then schedules renewal of the certificate, to occur at a specific time or periodically, depending on configuration. ACOS bases the new expiration date on the later of the expiration dates of the enrolled certificate and the CA certificate.
8. Rotate and store files. After certificate renewal, the old certificate and key files are still stored for any future reference. Old files are rotated, and new files replace the existing files.
9. This step ensures that there is no need to change the configuration for applications that use the SCEP certificates, because a valid certificate with the

correct name is always stored in the same location. The same applies for private keys as well. ACOS stores up to 4 old certificate and key files for each SCEP certificate.

Configuring SCEP Certificate

To configure SCEP using the CLI:

1. Use the `pki scep-cert` command to create the certificate and change the CLI to edit it.
2. Use the `url` command to specify the location of the ES. The user is the admin name required by the ES to accept the request.

The user is the admin name required by the ES to accept the request. The host is the ES IP address or hostname. The file is the path and filename for the SCEP process on the ES. Example:

url http://192.168.230.101/certsrv/mscep/mscep.dll

3. Specify the password for the certificate. ACOS includes this password in enrollment and renewal requests for the certificate.
4. (Optional) Configure additional parameters.
 - SCEP certificates have the following default settings:
 - Interval – 5 seconds
 - Log level – 1
 - Maximum poll time – 180 seconds
 - Method – GET
 - The other parameters are not set by default.
5. Use the `enroll` command to begin the enrollment process for the certificate.

Configuration Example

The following commands configure an the inside ACOS device in an SSLi deployment. The wildcard VIP on this device receives SSL-encrypted traffic from inside users, and decrypts the traffic before sending it to the security stack.

The deployment uses a certificate administered by an SCEP ES. Based on the configuration, ACOS automatically renews the certificate on a monthly basis.

For brevity, this example shows only the inside device, where SCEP configuration occurs, and uses only one certificate. The certificate is used both as the root certificate and as a forward-proxy certificate, which uses SNI support.

On the outside device, the only required command related to SSLi is forward-proxy-enable, to enable support for the SSLi feature on the device.

- The following commands enroll the certificate. You need to enroll each certificate only once. After a certificate is enrolled, ACOS uses SCEP to administer the certificate. This includes renewing the certificate before it expires. You do not need to manually administer the certificates after you enroll them.

```
ACOS(config)# pki scep-cert mycert  
ACOS(config-scep cert:mycert)# url  
http://192.168.230.101/certsrv/mscep/mscep.dll  
ACOS(config-scep cert:mycert)# password sample_password  
ACOS(config-scep cert:mycert)# renew-every month 1
```

- The following commands configure the client-SSL template:

```
ACOS(config)# slb template client-ssl ssl_int  
ACOS(config-client ssl)# certificate mycert key mycert  
ACOS(config-client ssl)# forward-proxy-enable  
ACOS(config-client ssl)# forward-proxy-ca-certificate Cert123.pem key  
key123
```

- The following command copies the SCEP certificate and key to a destination certificate file along with the rotation number of the SCEP generated certificate file.

```
ACOS(config)# pki copy-cert mycert rotation 2 testcert
```

- The following shows the configuration the wildcard VIP. This includes configuration of the other resources, in addition to the client-SSL template, that are required by the wildcard VIP: an ACL that matches on the inside clients, the real server configuration, and the service group.

```
access-list 101 permit ip any 10.2.2.0 0.0.0.255 log  
!  
slb server rs1 10.3.3.1
```

```
health-check-disable
port 443 tcp
    health-check-disable
!
slb service-group sg1-tcp tcp
    member rs1:443
!
slb virtual-server vs1-v4 0.0.0.0 acl 101
    extended-stats
    port 8080 http
        service-group sg1-tcp
        template client-ssl ssl_int
        no-dest-nat port-translation
!
```

- The following shows the configuration the wildcard VIP. This includes configuration of the other resources, in addition to the client-SSL template, that are required by the wildcard VIP: an ACL that matches on the inside clients, the real server configuration, and the service group.

```
access-list 101 permit ip any 10.2.2.0 0.0.0.255 log
!
slb server rs1 10.3.3.1
    health-check-disable
    port 443 tcp
        health-check-disable
!
slb service-group sg1-tcp tcp
    member rs1:443
!
slb virtual-server vs1-v4 0.0.0.0 acl 101
    extended-stats
    port 8080 http
        service-group sg1-tcp
        template client-ssl ssl_int
        no-dest-nat port-translation
!
```


- The following commands show information about the certificate:

```

ACOS(config)# show pki cert
Name: mycert Type: certificate/key Expiration: Dec 8 22:23:48 2014
GMT [Expired, Bound] SCEP Enrolled

ACOS(config)# show pki cert mycert
Certificate:
  Data:
    Version: 3 (0x2)
    Serial Number:
      1d:5b:42:30:00:00:00:00:24:8f
    Signature Algorithm: sha1WithRSAEncryption
    Issuer: DC=com, DC=a10lab, CN=AD03-CA
    Validity
      Not Before: Dec 8 18:23:48 2014 GMT
      Not After : Dec 8 22:23:48 2014 GMT
    Subject: C=CH, O=Linux strongSwan, CN=AX1030
    X509v3 extensions:
      X509v3 Subject Key Identifier:

DA:53:59:9C:EC:52:E3:58:6C:E5:84:11:E7:5C:F4:C9:FC:59:6B:A3
      X509v3 Authority Key Identifier:

keyid:06:18:97:1C:58:B4:E4:95:5F:61:61:5D:DB:9C:1B:85:39:48:87:37

      X509v3 CRL Distribution Points:
        URI:ldap:///CN=AD03-
CA,CN=AD03,CN=CDP,CN=Public%20Key%20Services,CN=Services,CN=Configurati
on,
DC=a10lab,DC=com?certificateRevocationList?base?objectClass=cRLDistribu
tionPoint

      Authority Information Access:
        CA Issuers - URI:ldap:///CN=AD03-
CA,CN=AIA,CN=Public%20Key%20Services,CN=Services,CN=Configuration,
DC=a10lab,DC=com?cACertificate?base?objectClass=certificationAuthority
        OCSP - URI:http://ad03.a10lab.com/ocsp

```

```
X509v3 Key Usage: critical
    Digital Signature, Key Encipherment
1.3.6.1.4.1.311.21.7:
    0-%+.....7.....E.....+.....Ks...M.....d...
X509v3 Extended Key Usage:
    1.3.6.1.5.5.8.2.2
    1.3.6.1.4.1.311.21.10:
    0.0
```

Configuring Automatic Certificate Management Environment (ACME) Certificates

ACME is used to obtain the certificates for websites (HTTPS). The purpose is to validate domain names for issuing certificates in the web PKI. This protocol is based on passing JSON-formatted messages over HTTPS and was designed by the Internet Security Research Group (ISRG) in RFC 8555 for their Let's Encrypt service. It also enables automating a few aspects of certificate management.

Using both Let's Encrypt and the ACME protocol, you can set up an HTTPS server and automatically obtain a browser-trusted certificate. Generally, ACME client runs as an agent on a web server. The ACME client is integrated into ACOS to support obtaining and renewing the certificate.

Consider the following key points:

- The domain is verified using challenge HTTP-01, which is used to provisioning an HTTP resource with a well-known URI. Hence, you need to first enable reply ACME HTTP-01 challenge option for the CA server. This option is allowed only on HTTP port 80.
- HTTP port 80 should not be blocked by the firewall. The virtual ports 80 replying ACME challenges must be up and running.
- The domain must be certificate's Common Name and the IP address must be mapped to the ACOS's virtual IP address.
- The CA server determines whether the ACME client has domain control. Hence, you must manually configure reverse proxy on the ACOS side.
- ACME challenge from the CA server goes into the data interface.

- ACME supports shared and L3V partitions. ACME protocol traffic follows the routing table rule and DNS setting on the shared partition. Hence, the network interfaces, routing rules, and DNS must be configured correctly in the shared partition, as the ACME certificate is created in the L3V partition.
- In aVCS with VRRP-A high availability deployment, aVCS master accepts the configuration, and the master syncs the configuration to the slave device(s). Data traffic is served by HA primary device(s), so HTTP-01 type challenge from the CA server is also served by HA primary device(s). If you do not want to utilise the default VRID(0), you can change it with the VRID option.

NOTE:

- Lets Encrypt or ACME feature is not supported in a VRRP-A standalone environment mode.
- The VIP must be publicly accessible for certificate validation. To configure the VIP IP as the same as the data interface on vThunder, use the `use-if-ip` option.

The following features are supported :

- Only the HTTP-01 method is supported for domain validation challenges.
- Supports aVCS with VRRP-A for high availability, but does not support VRRP-A standalone deployment.
- [External Account Binding \(EAB\)](#) is supported to bind an external data source.

The following topics are covered:

Enrollment and Renewal Process	535
Configuring ACME Certificate	537
Configuring External Account Binding	538
Configuration Examples	538

Enrollment and Renewal Process

After you configure an ACME certificate for enrollment, ACOS performs the following steps:

- Generates an account key and registers this account with the CA server.
- Generates a domain private key and RSA or EC key with the specified key-length for the certificate.
- ACOS proves the CA server that the user domain is in control. If the domain is already verified, ACOS skips this step. If the domain is not verified yet, ACOS deploys the challenge and triggers the CA server to start the verification process.
 - Certificate Issuance - Once the CA verification succeeds, the ACOS account key pair is authorized, requested, renewed, to create/send CSR and sign them with the authorized key pair. Then, ACOS downloads certificates for the domain from the CA server.
 - Store the certificate. After successful verification of the response from the CA, ACOS accepts the certificate and stores it.
 - Notify application layer - The client-SSL template would accept the ACME certificate. ACOS notifies the application that the certificate is ready for use.
 - Schedule renewal - ACOS handles the automatic renewal of the certificate when it is about to expire. ACOS checks the expiration date, depending on the periodic renewal or a specific time before the certificate expires configurations.
 - Rotate and store files - After certificate renewal, the old certificate and key files are still stored for any future reference. Old files are rotated and the new file replaces the existing files.

NOTE:

- After enrollment, the account-email and domain are already registered with CA. Hence, account-email and domain cannot be changed after certificate is enrolled. To change these, you must remove acme-cert and re-enroll.
 - When the ACME enrollment or renewal fails, the scheduled renewal time continues as planned, and the next renewal time will be scheduled.
-

You can configure different ACME clients (`agent-v1` or `agent-v2`) to perform the ACME enrollment or renewal. The default agent is `agent-v1`.

To configure `agent-v1` as the ACME client to perform ACME enrollment, use the following command:

```
ACOS(config-acme cert:cert1)# acme-client agent-v1
```

You can also configure the number of concurrent ACME processes. The ACME concurrent process count indicates the number of ACME enrollment or renewal operations that can run concurrently.

The default and the maximum number of ACME concurrent processes depend on the number of control CPUs. To set the number of ACME concurrent processes, use the following command:

```
ACOS(config)# pki acme-concurrent-process-count <num>
```

If the number of control CPUs is greater than 12, the default and the maximum numbers of ACME concurrent processes are limited to the number of 12 control CPUs. For more information, see *Command Line Interface Reference*.

Table 11 : Default and Maximum Numbers of ACME Concurrent Process

No. of control CPUs	Default # of concurrent process	Maximum # of concurrent process
1	5	7
2	9	11
3	13	15
4	17	19
5	21	23
6	25	27
7	29	31
8	33	35
9	37	39
10	41	43
11	45	47
12	49	51
# > 12	49	51

Configuring ACME Certificate

To configure ACME using the CLI:

1. Use the `reply-acme-challenge` option under HTTP virtual port.
2. Use the `pki acme-cert` command to create the certificate and change the CLI to edit it.
3. Use the `account-email`, `cert-type`, `domain`, and `url` command.
4. (Optional) Configure additional parameters.
 - Renew Intervals
 - Log level
 - Staging URL
 - SAN domain
 - VRID
5. Use the `enroll` command to begin the enrollment process for the certificate.

Configuring External Account Binding

External Account Bindings allows you to associate your ACME account with an external account such as a CA custom database.

To enable ACME account binding:

1. The CA operating ACME server needs to provide ACME client with the EAB credentials i.e., a MAC key and a key identifier, using some mechanism outside of ACME.
2. Configure the EAB credentials using `pki acme-cert` command.
 - `eab-hmac-key` - Specify the HMAC key for ACME EAB. The MAC key should be in base64url-encoded format.
 - `eab-key-id` - Specify the key identifier for ACME EAB. The key identifier must be an ASCII string.

Configuration Examples

- The following commands enroll the certificate with the Let's encrypt CA. You need to enroll each certificate only once. After a certificate is enrolled, ACOS uses ACME to administer the certificate. This includes renewing the certificate before it expires. You do not need to manually administer the certificates after you enroll

them.

```
ACOS(config)# pki acme-cert test
ACOS(config-acme cert:test)# account-email test@url.com
ACOS(config-acme cert:test)# cert-type rsa
ACOS(config-acme cert:test)# domain test.com
ACOS(config-acme cert:test)# enroll
ACOS(config-acme cert:test)# exit
```

- The following commands show information about the ACME certificate. You can view both the status and the detail information of the ACME certificate. Also, **show pki acme-cert log <cert-name>** can be used to display the detailed log of the ACME protocol happened during the ACME cert registration or update process.

```
ACOS(config)# show pki acme-cert status
Certificate name: test status: SUCCESS
    Renew every 2 minutes
    rotated files: 4

ACOS(config)# show pki cert acme-test-cert detail
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 1770931951 (0x698e46ef)
    Signature Algorithm: sha256WithRSAEncryption
    Issuer: CN=aws2020.ddns.net
    Validity
        Not Before: Oct 15 02:30:26 2019 GMT → issued time
        Not After : Oct 14 02:30:26 2021 GMT → cert validity time
    Subject: CN=hku_ra, O=MyOrganization, C=SE
```

- The following command clears the ACME certificate logs and debug information.

```
ACOS(config)# clear pki acme-cert log acme-test-cert
ACOS(config)#
```

SSL Offload and SSL Proxy

This section describes how to configure optimization of Secure Sockets Layer (SSL).

The following topics are covered:

SSL Offload	541
SSL Proxy	546
Overview	549

SSL Offload

ACOS SSL Insight (SSLi) provides SSL security services through its transparent forward proxy traffic inspection application.

The following topics are covered:

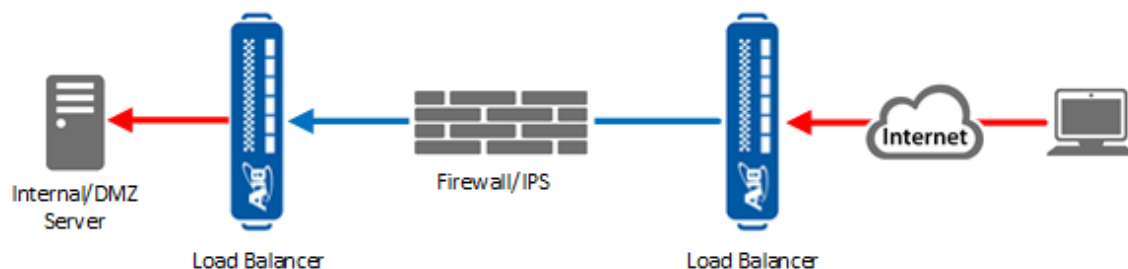
Overview	541
Configuring SSL offload	541

Overview

In SSL offload, HTTPS load balancing (Layer 7 SLB) can be combined with optional HTTP packet inspection and header manipulation.

SSL offload configures the HTTPS virtual port type to enable the SSL handshake and optionally configures the HTTP template to enable packet inspection and header manipulation.

Figure 55 : SSL Offload (Load Balancing HTTPS Traffic)



Configuring SSL offload

This section explains configuring SSL offload.

Configuring the Client SSL Template (CLI Example)

NOTE: The procedures importing or creating certificates is the same whether they are used in an SSL proxy deployment or an SSL offload deployment. Similarly, the procedure for configuring a client SSL template is the same for SSL proxy and SSL offload.

1. Import or create a certificate and its key to use for TLS sessions with clients. The configuration example in this section uses an imported SSL certificate and private key.

```
ACOS# import cert sslcert1.crt ftp:
Address or name of remote host []?1.1.1.2
User name []?Admin-15
Password []?*****
File name [/]?sslcert1.crt
ACOS# import key sslcertkey.pem ftp:
Address or name of remote host []?1.1.1.2
User name []?Admin-15
Password []?*****
File name [/]?sslcertkey.pem
```

2. Configure a client SSL template and bind the SSL certificate and private key to it.

ACOS uses SSL certificates with a private key to create proxied signed certificates for handshaking with SSL clients. SSL clients are configured to use a private CA that verifies the proxied certificate's validity.

```
ACOS(config)# slb template client-ssl sslcert-tmplt
ACOS(config-client ssl)# certificate sslcert.crt key sslcertkey.pem
ACOS(config-client ssl)# exit
```

Configuring SSL Offload (CLI Example)

1. Configure the client SSL template using the procedure described in [CLI Configuration](#).
2. Configure the real servers for the TCP service:

```
ACOS(config)# slb server HTTPS1 10.5.5.2
ACOS(config-real server)# port 80 tcp
```

```
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
ACOS(config)# slb server HTTPS2 10.5.5.3
ACOS(config-real server)# port 80 tcp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
```

3. Configures a service group for the HTTPS servers:

```
ACOS(config)# slb service-group HTTPS_servers tcp
ACOS(config-slb svc group)# member HTTPS1 80
ACOS(config-slb svc group-member:80)# exit
ACOS(config-slb svc group)# member HTTPS2 80
ACOS(config-slb svc group-member:80)# exit
ACOS(config-slb svc group)# exit
```

4. Configure a virtual server and add a virtual port that has the service type `https`. Bind the service-group to the virtual port and to the HTTP template (if configured) and client-SSL template.

The SSL offload feature is enabled by the `https` option of the `port` command.

```
ACOS(config)# slb virtual-server v1 10.6.6.6
ACOS(config-slb vserver)# port 443 https
ACOS(config-slb vserver-vport)# service-group HTTPS_servers
ACOS(config-slb vserver-vport)# template client-ssl sslcert-tmpl
```

5. In this example, traffic between the servers and ACOS is not encrypted.
- If traffic between the servers and ACOS also is encrypted, you also need to configure a server-SSL template and bind it to the virtual port. In configurations that use both client-SSL and server-SSL, use the HTTPS/SSL port number in the real server configuration.
 - If only client-SSL is used, use the HTTP port number in the real server configuration. Use the HTTPS/SSL port number in the virtual server configuration.
 - If you configure server-SSL without client-SSL, the virtual port service type must be HTTP, not HTTPS.
6. For configuring HTTP packet and inspection, see the “HTTP Options for SLB” section of the *Application Delivery Controller Guide*.

GUI Configuration

- To import the SSL certificate:
 1. Navigate to **ADC >> SSL Management >> SSL Certificates >> Import**.
 2. Enter the file name, the category of the import (Certificate), the location of the import (Local, Remote, or Text; if Remote or Text is selected the appropriate fields will pop up to indicate IP or text content), certificate format, and the certificate source for file search.
 3. Click **Import**.
- To import the SSL key:
 1. Navigate to **ADC >> SSL Management >> SSL Certificates >> Import**.
 2. Enter the file name, the category of the import (Key), the location of the import (Local, Remote, or Text), and the key source for file search.
 3. Click **Import**.
- To create the client SSL template using the imported certificate and key:
 1. Navigate to **ADC >> Templates >> SSL**.
 2. Select **Create** to pull the drop-down menu, and click **Client SSL**.
 3. In the **Name** field, specify a name for the Client SSL template.
 4. In the **Server Certificate** field, select an imported or configured SSL certificate, and in the **Server Private Key** field, select an imported or configured key. ACOS uses the SSL certificates with the private key to create proxied signed certificates for handshaking with SSL clients. The SSL clients are configured to use a private CA that verifies the proxied certificate's validity.
 5. Click **OK** to finish.
- Configure the real servers for the TCP service:
 1. Navigate to **ADC >> SLB >> Servers** tab.
 2. Click **Create**.
 3. In the **Name** field, enter a name for the real server.
 4. Select the address **Type** radio button: IPv4, IPv6, or FQDN.
 5. Enter the IP Address or FQDN hostname.

6. In the Port section, click **Create**.
 7. In the **Port Number** field, enter the port number.
 8. Click the **Protocol** drop-down list, and select **TCP**, if not already selected.
 9. If your configuration will use a health check, select the **Default** radio button to use the default health check, or select the **Monitor** radio button and select the desired health monitor from the menu.
 10. Click **Create**. The port appears in the Port list.
 11. Click **Update**. The server appears in the real server table.
 12. Repeat for each real server.
- Configures a service group for the HTTPS servers:
 1. Select the **Service Groups** tab from the menu bar.
 2. Click **Create**. The Create Service Group page appears.
 3. In the **Name** field, enter a name for the service group.
 4. In the **Protocol** drop-down list, select **TCP**, if not already selected.
 5. In the Member section, click **Create**, and then click the **Server** drop-down list and select a server.
 6. In the **Port** field, enter the service port number.
 7. Click **Create**. The member server appears in the list.
 8. Repeat step e through step g for each server that will be added to the service group.
 9. Click **Create**. The new service group appears in the service group table.
 - Configure a virtual server and add a virtual port that has the service type **HTTPS**. Bind the service-group to the virtual port and to the **Template HTTP** (if configured) and **Template Client-SSL**.
 1. Navigate to **ADC >> SLB >> Virtual Servers** tab.
 2. Click **Create**.
 3. In the **Name** field, enter a name for the virtual server.
 4. Select the **Address Type** radio button (IPv4 or IPv6), and then enter the VIP address. This is the IP address to which client requests will be sent.

5. In the Virtual Port section, click **Create**.
6. From the page that appears, click the **Protocol** drop-down menu and select **HTTPS**.
7. The SSL offload feature is enabled by the **HTTPS** option.
8. In the **Port** field, enter the service port number.
9. Click the **Service Group** drop-down menu and select the service group.
10. Click **Templates** to expand the template configuration options.
11. Click the **Template HTTP** drop-down list, select the template you configured earlier.
12. Click the **Template Client-SSL** drop-down list, select the desired template.
13. Click **Create**. The port appears in the Port list for the virtual server.
14. Click **Update**. The virtual server appears in the virtual server table.

SSL Proxy

Overview

In *SSL proxy*, the ACOS device acts as a Layer 4 SSL proxy for TCP services such as POPs, SMTPS, IMAPS, and LDAPS. It combines TCP load balancing (Layer 4 SLB) with these proxy services.

Both types of SSL optimization perform SSL handshakes, as well as encryption / decryption. SSL certificates and keys are required. You can import the certificates and keys or create them on the ACOS device. Additionally, ACOS also provides support for ECDHE/DHE ciphers, including ECDHE-RSA ciphers, DHE-RSA ciphers, ECDHE-ECDSA ciphers, and GCM & SHA384. This feature also allows for the configuration of EC and DH parameters, EC Curve selection, the importing/verification of EC Keys for ECDSA ciphers, and support for TLS1.0/TLS1.1.

Configuring SSL Proxy

CLI Configuration

1. Configure the client SSL template using the procedure described in [CLI Configuration](#).

2. Configure the real servers for the TCP service:

The following commands configure proxy SSL for POPS. The same commands can be used to configure SSL proxy for other TCP services. In each case, the feature is enabled by the `ssl-proxy` option of the `port` command at the virtual server configuration level of the CLI.

```
ACOS(config)# slb server POP1 10.5.5.2
ACOS(config-real server)# port 110 tcp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
ACOS(config)# slb server POP2 10.5.5.3
ACOS(config-real server)# port 110 tcp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
```

3. The following commands configure a service group for the POP servers:

```
ACOS(config)# slb service-group POP_servers tcp
ACOS(config-slb svc group)# member POP1 110
ACOS(config-slb svc group-member:110)# exit
ACOS(config-slb svc group)# member POP2 110
ACOS(config-slb svc group-member:110)# exit
ACOS(config-slb svc group)# exit
```

4. These commands configure a virtual server (VIP) which proxies for the service POP server (port 110):
5. The following commands configure the VIP to which clients will send POPS traffic (that is, port 110):

```
ACOS(config)# slb virtual-server v1 10.6.6.6
ACOS(config-slb vserver)# port 110 ssl-proxy
ACOS(config-slb vserver-vport)# service-group POP_servers
ACOS(config-slb vserver-vport)# template client-ssl sslcert-tmpl
```

NOTE: Consider using the `use-client-sni` command in configurations where the SSL server hosts multiple services.

GUI Configuration

1. Configure the client SSL template using the procedure described in [CLI Configuration](#).
2. Configure the real servers for the TCP service:
 - a. Navigate to **ADC >> SLB >> Servers** tab.
 - b. Click **Create**.
 - c. In the **Name** field, enter a name for the real server.
 - d. Select the address **Type** radio button: IPv4, IPv6, or FQDN.
 - e. Enter the IP Address or FQDN hostname.
 - f. In the Port section, click **Create**.
 - g. In the **Port Number** field, enter the port number.
 - h. Click the **Protocol** drop-down list, and select **TCP**, if not already selected.
 - i. If your configuration will use a health check, select the **Default** radio button to use the default health check, or select the **Monitor** radio button and select the desired health monitor from the menu.
 - j. Click **Create**. The port appears in the Port list.
 - k. Click **Update**. The server appears in the real server table.
 - l. Repeat for each real server.
3. Configure a service group for the service corresponding the **Port Number**:
 - a. Select the **Service Groups** tab from the menu bar.
 - b. Click **Create**. The Create Service Group page appears.
 - c. In the **Name** field, enter a name for the service group.
 - d. Click the **Protocol** drop-down list and select **TCP**, if not already selected.
 - e. Select the **Health Monitor** from the drop-down, if your configuration will use one.

- f. In the **Member** section, click **Create**, and then click the **Server** drop-down list and select a server.
 - g. In the **Port** field, enter the service port number for this server.
 - h. Click **Create**. The server appears in the list.
 - i. Repeat for each server.
 - j. Click **Update**. The service group appears in the service group table.
4. Configure a virtual server (VIP) which proxies for the service corresponding the **Port Number**:
 - a. Navigate to **ADC >> SLB >> Virtual Servers** tab.
 - b. Click **Create**.
 - c. In the **Name** field, enter a name for the virtual server.
 - d. Select the **Address Type** radio button (IPv4 or IPv6), and then enter the VIP address. This is the IP address where client requests will be sent.
 - e. In the Virtual Port section, click **Create**. The Create Virtual Port page appears.
 - f. Click the **Protocol** drop-down menu and select **SSL-Proxy**.
 - g. In the **Port** field, enter the service port number.
 - h. Click the **Service Group** drop-down menu and select the service group.
 - i. Click **Templates** to expand the template configuration options.
 - j. Click the **Template Client-SSL** drop-down list, and select the desired template.
 - k. Click **Create**. The SSL proxy port appears in the port list for the virtual server.
 - l. Click **Update**. The virtual server appears in the virtual server table.

Overview

Both types of SSL optimization perform SSL handshakes, as well as encryption / decryption. SSL certificates and keys are required. You can import the certificates and keys or create them on the ACOS device. ACOS also provides support for ECDHE/DHE ciphers, including ECDHE-RSA ciphers, DHE-RSA ciphers, ECDHE-ECDSA ciphers, and GCM & SHA384. This feature also allows for the configuration of EC and DH

parameters, EC Curve selection, the importing/verification of EC Keys for ECDSA ciphers, and support for TLS1.0/TLS1.1.

When configuring the client or server SSL template, users will also have the option to configure ECDHE/DHE ciphers for enhanced encryption and SSL processing. Software-based SSL processing supports ECDHE/DHE ciphers for all vThunder platforms.

Additionally, ACOS features enhanced selection of cipher support based on priorities assigned to ECDHE and DHE cipher templates configured on the ACOS device:

- When processing an SSL handshake, if a template is configured a template for both ECDHE and DHE with same priority levels, priority is given to ECDHE over DHE to optimize CPU usage on the ACOS device. DHE ciphers are as the lowest priority if there are other supported ciphers in the client-hello message. But if the highest priority is configured for a DHE cipher, the ACOS device honors that.
- If the customer has a cipher template where no priority is specified, the ACOS device gives ECDHE a higher priority by default. However, it is strongly recommended that the priority is not unspecified.
- PFS ciphers on FIPS platform is not supported; ciphers for server-side SSL are supported in software.

Configuring SSL Ciphers

ECDHE/DHE ciphers can be supported in TLS1.0/TLS1.1. GCM related ciphers are only supported in TLS1.2 For a list of supported ciphers, refer to the **A10 SSL Cipher Suites List** file located on the [A10 Networks Support Portal](#). Log in using your GLM credentials and browse to **App/Tech Notes**.

1. To use GCM related ciphers in Server SSL templates, you will need to specify TLS version 1.2:

```
ACOS(config)# slb template server-ssl SERVER-1
ACOS(config-server ssl)# version ?
<30-33> TLS/SSL version: 30-SSLv3.0, 31-TLSv1.0, 32-TLSv1.1 and 33-
TLSv1.2
```

2. You can now specify the Elliptic Curve Name in Client SSL templates:

```
ACOS(config)# slb template client-ssl CLIENT-1
ACOS(config-client ssl)# ec-name ?
    secp256r1    X9_62_prime256v1
    secp384r1    secp384r1
    secp521r1    secp521r1
    x25519       x25519
```

3. If no EC name is specified, ACOS will pick the first one that can be supported in ACOS from the client EC list. If an EC name(s) is specified, ACOS will pick the first one that can be supported by the client.
4. You can now specify DH parameters in Client SSL templates:

```
ACOS(config-client ssl)# dh-param ?
    1024
    1024-dsa
    2048
```

The command shown above specifies DH key length. By default, the length is 1024. ACOS does not have configurable DH parameters in Server SSL templates as the client uses the real server's DH parameters.

5. You can now specify the EC name in Server SSL templates. The command is the same as when you specify the EC name in Client SSL templates:

```
ACOS(config-server ssl)# ec-name ?
    secp256r1    X9_62_prime256v1
    secp384r1    secp384r1
    secp521r1    secp521r1
    x25519       x25519
```

By default, if no EC name is specified, ACOS will send all supported EC names to the server. If an EC name(s) is selected, ACOS will send the specified EC name(s) to the server.

Configuration Examples

SSL Cipher Configuration Example

Below is an example for ACOS devices:

```
slb template client-ssl clientssl
```

```
certificate cert key key  
ec-name secp384r1  
cipher TLS1_ECDHE_RSA_AES_128_SHA256  
slb template server-ssl serverssl  
cipher TLS1_ECDHE_RSA_AES_128_SHA256
```

Secure FTP Proxy

The following topics are covered:

Overview	554
Configuring SSL Offload for Secure FTPS Traffic	557

Overview

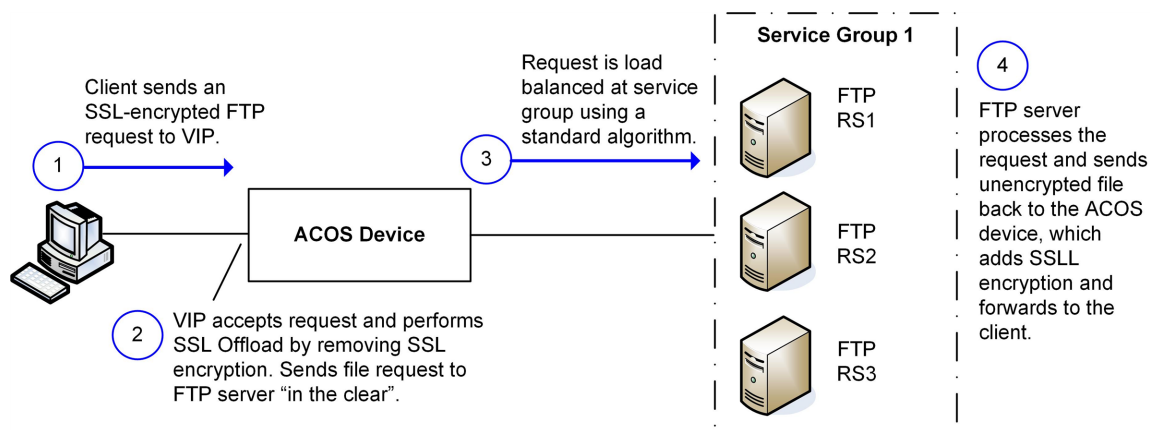
ACOS provides a virtual port type, FTP-proxy. You can use an FTP-proxy virtual port to load balance secure SSL/TLS traffic or clear-text FTP traffic for clients.

While previous releases supported SSL offload for HTTPS traffic, this release supports similar functionality for encrypted FTPS traffic. Explicit FTPS traffic is an extension to the FTP protocol which allows the FTP client to request that the session be encrypted with SSL/TLS.

Since the connection between client and ACOS device is secure, the ACOS device provides SSL proxy services for the FTP traffic during negotiation of the secured session and acts as a proxy for backend FTP servers.

By encrypting / decrypting traffic to and from the FTP servers, the ACOS device handles this CPU-intensive task, thus sparing the FTP servers and enabling them to respond more quickly to client requests.

Figure 56 : Secure FTP Proxy



- In the current release, Secure FTP is only supported for Explicit FTPS transactions.
- In passive FTP mode, the server specifies which server-side port the client should connect to and the client initiates the connection. This is in contrast to active FTP mode, where the client specifies which port it has opened up for the data channel, and the server initiates the connection.

- After receiving the AUTH TLS command from the FTP client, ACOS expects to receive an SSL handshake for the control channel, and expects the rest of the traffic from client to be encrypted.
- ACOS supports the FTP clear command channel (CCC) command. Meaning that after ACOS receives the CCC command from the FTP client, the encryption is removed and packets are sent as clear text.
- The data channel may or may not be encrypted, depending on which PROT command is sent from the FTP client. The PROT P command indicates that the data is encrypted, whereas PROT C indicates the data is not encrypted.

For more information about SSL offload, refer to [SSL Offload and SSL Proxy](#).

FTP Proxy Traffic Flow

1. A client sends an encrypted Explicit FTPS request to the ACOS VIP.
2. The ACOS VIP acts as a proxy for the backend FTP real server, and performs SSL offload by removing the TLS encryption.
3. The client's unencrypted FTP request is load balanced among the FTP servers using a standard load balancing algorithm.
4. The FTP server receives the file request and responds by sending the requested file back to the ACOS device "in the clear". The ACOS device re-encrypts the FTP traffic from the server, creating FTPS traffic, and sends the encrypted FTPS file to the client.

Explicit Passive FTP Proxy Session Example

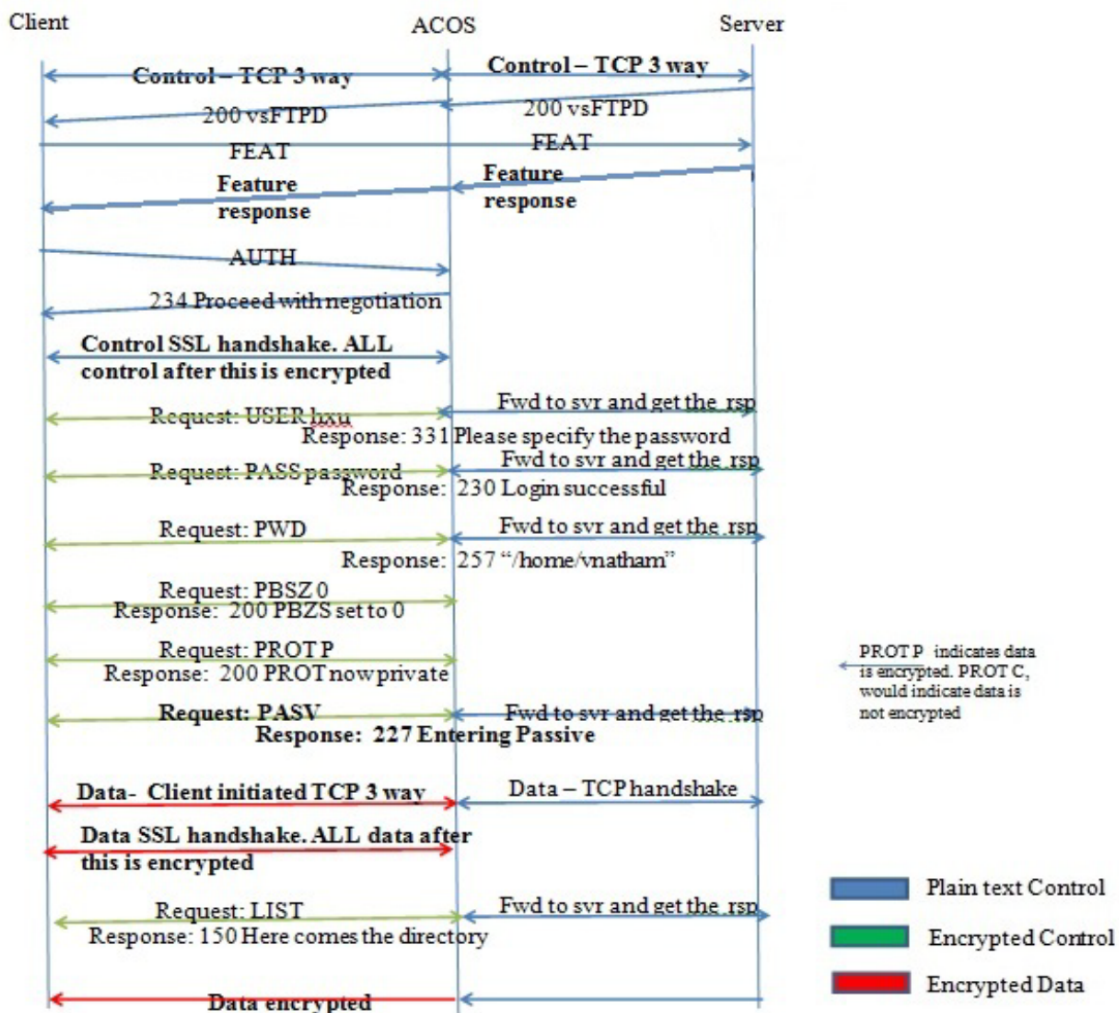
[Explicit passive FTP proxy session](#) shows a sample of an explicit passive FTP proxy session.

The diagram shows traffic flowing back and forth between a client and an FTP server with an ACOS device in the middle acting as a proxy.

- A standard 3-way TCP handshake occurs on both sides of the ACOS device, and this traffic is unencrypted, which is represented in the diagram with blue lines.

- Next, an SSL handshake occurs between the client and the ACOS device. Once the SSL handshake is finished, the rest of the FTP control traffic is encrypted between the client and the ACOS device (as shown with the green lines).
- The PROT P command can indicate the data channel will be encrypted, as shown with the red lines.
- Communication between the ACOS device and the FTP server is never encrypted in this example.

Figure 57 : Explicit passive FTP proxy session



Configuring SSL Offload for Secure FTPS Traffic

Configuring SSL Offload for Secure FTPS Traffic (Procedure)

1. Configure client SSL. (See [CLI Configuration](#)).
2. Configure the real FTP servers.
3. Configure a service group for the servers and add them to the group.
4. Configure client-SSL template options.
5. Configure a virtual server and add a virtual port that has the service type ftp-proxy. Bind the service-group to the virtual port and to the client-SSL template.

Since traffic between FTP servers and ACOS device is not encrypted, a server-SSL template is not required.

GUI Configuration

To configure the real FTP server(s):

1. Navigate to **ADC >> SLB >> Servers** tab.
2. Click **Create**.
3. In the **Name** field, enter a name for the real server.
4. Select the address **Type** radio button: IPv4, IPv6, or FQDN.
5. Enter the IP Address or FQDN hostname.
6. In the Port section, click **Create**.
7. In the **Port Number** field, enter the port number (for example, Port 21).
8. Click the **Protocol** drop-down list, and select **TCP**, if not already selected.
9. If your configuration requires health checks, select the **Default** radio button to use the default health check or select the **Monitor** radio button and select a desired health monitor from the menu.
10. Click **Create** to add this port to the real FTP server.
11. Click **Update**. The server appears in the real server table.
12. Repeat for each FTP server.

To configure a service group and add the real servers:

1. Select the **Service Groups** tab from the menu bar.
2. Click **Create**. The Create Service Group page appears.
3. In the **Name** field, enter a name for the service group.
4. In the **Protocol** drop-down list, select **TCP**, if not already selected.
5. Select the **Health Monitor** from the drop-down menu, if your configuration will use one.
6. In the Member section, click **Create**, and then click the **Server** drop-down list and select an FTP server.
7. In the **Port** field, enter the service port number.
8. Click **Create**. The server appears in the list.
9. Repeat these steps for each FTP server that will be added to the service group.
10. Click **Create**. The new service group appears in the service group table.

To configure a virtual server for SSL offload for FTP traffic:

1. Navigate to **ADC >> SLB >> Virtual Servers** tab.
2. Click **Create**.
3. In the **Name** field, enter a name for the virtual server.
4. Select the **Address Type** radio button (IPv4 or IPv6), and then enter the VIP address. This is the IP address where client requests are sent.
5. In the Virtual Port section, click **Create**.
6. From the page that appears, click the **Protocol** drop-down menu and select **FTP-Proxy**.
7. In the **Port** field, enter the service port number (default port number for FTP is 21).
8. Click the **Service Group** drop-down menu and select the service group.
9. Click **Templates** to expand the template configuration options.
10. Click the **Template Client-SSL** drop-down list, select the desired template.

11. Click **Create**. The FTP-Proxy port appears in the port list for the virtual server.
12. Click **Update**. The virtual server appears in the virtual server table.

CLI Configuration

1. To configure a real FTP server, use the `slb server` command, then enter the `port` command at the configuration level for the real server.

```
ACOS(config)# slb template client-ssl CLIENT-1
ACOS(config)# slb server SERVER-1 10.1.1.1
ACOS(config-real server)# port 80 tcp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
```

2. To configure a service group for real FTP servers and add them to the group, use the `slb service-group` command, then enter the `member` command at the configuration level for the service group.

```
ACOS(config)# slb service-group GROUP-1 tcp
ACOS(config-slb svc group)# member SERVER-1 80
ACOS(config-slb svc group-member:80)# exit
ACOS(config-slb svc group)# exit
```

3. To configure a virtual server and FTP-Proxy virtual port, use the `slb virtual-server` command and enter `service-group` and `template client-ssl` commands at the configuration level for the virtual port to bind the port to the service group and the application templates. See [CLI Configuration](#) for instructions on configuring a client-SSL template.

```
ACOS(config)# slb virtual-server VIP-1 10.15.1.1
ACOS(config-slb vserver)# port 80 ftp-proxy
ACOS(config-slb vserver-vport)# service-group GROUP-1
ACOS(config-slb vserver-vport)# template client-ssl sslcert-tmpl
ACOS(config-slb vserver-vport)# exit
ACOS(config-slb vserver)# exit
```

Show and Clear SLB FTP-Proxy Commands

- The `show slb ftp-proxy` command displays ftp proxy output.
- The `clear slb ftp-proxy` command clears the counters that appear in `show slb ftp-proxy` output.

ALG Protocol FWLB Support for FTP and SIP

The following topics are covered:

Overview	562
Topologies	564
Persistent Sessions	567
Configuring ALG FWLB	570

Overview

In simple Firewall Load Balancing (FWLB) deployments, the ACOS device does not support the ability to load balance Application Layer Gateway (ALG) protocols, which have multiple connections that can originate from either side of the firewall deployment. This lack of predictability that occurs with ALG protocols can result in the protocol's control connection and data connection being sent to different firewalls, which can cause the application to stop working.

To handle some of the more complex ALG protocols, you can configure the ACOS device to load balance ALG protocols, such as FTP and SIP, through a firewall deployment consisting of paired firewalls through the use of persistent session templates.

When dealing with ALG protocols such as FTP and SIP, sessions can originate from either side of the firewalls. This unpredictable quality can pose a challenge to the load balancer(s) inside the FWLB deployment when they are tasked with setting up the persistent sessions that these protocols require.

This ALG protocol FWLB feature resolves such issues with session persistence across a firewall deployment for FTP and SIP traffic.

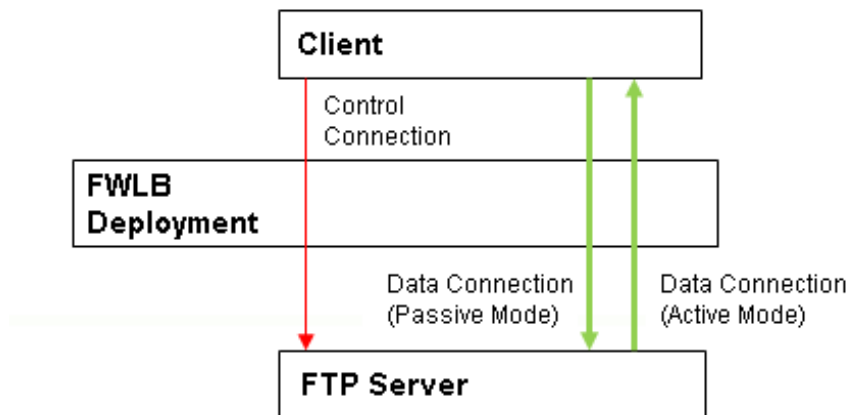
FTP Support

[Figure 58](#) shows FTP traffic passing back and forth between a client and an FTP server. ACOS uses standard SLB server selection methods to choose a firewall that will handle the client's traffic.

The FTP protocol requires two separate sessions, or connections, which are represented in [Figure 58](#) with red and green arrows:

- The red arrow represents the control connection.
- The green arrows represent the data connections (or “out of band” connections).

Figure 58 : FTP Traffic Flows



The control connection (red arrow) is usually initiated by the client, while the data connections (green arrows) can be initiated from either side of the FWLB deployment and can be initiated by either the client or the FTP server.

If the client initiates the data connection, then the FTP transaction is said to be in “passive” mode. This is because the FTP server remains passive. However, if the data connection is initiated by the FTP server, then the FTP connection is said to be in “active” mode because the FTP server is taking action.

SIP Support

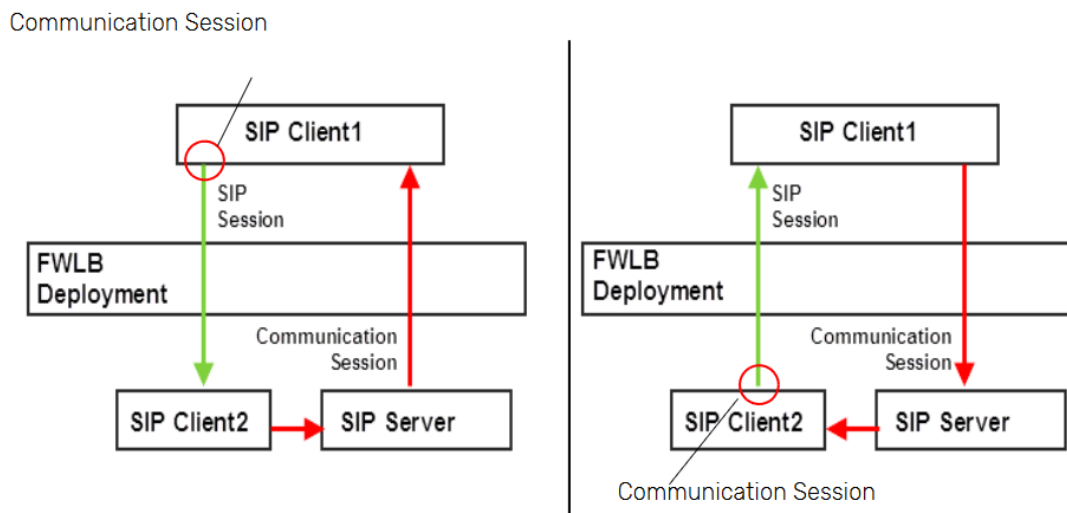
As is the case with FTP, Session Initiation Protocol (SIP) poses unique challenges for the ACOS load balancers that are attempting to create persistent sessions across a pair of firewalls in an FWLB deployment.

[Figure 59](#) shows two separate SIP transactions side by side. Both scenarios involve a SIP server and two or more SIP clients.

The SIP protocol requires two separate sessions, which the figure represents with red and green arrows:

- The green arrow represents the Communication Session.
- The red arrows represent the SIP Sessions.

Figure 59 : SIP traffic originating from both sides of FWLB deployment



- The initial communication session is launched from a SIP client (as opposed to the SIP server). This communication session can be launched from either side of the FWLB deployment.
- In the leftmost scenario shown in [Figure 59](#), the communication session originates from SIP client 1, while the scenario on the right shows the communication session originating from SIP client 2, (in which case SIP client 2 is on the same side of the firewall as the SIP server).
- Once the communication session is established between the SIP server and client, then the SIP clients can communicate through a separate SIP session.
- The load balancers in the FWLB deployment must be able to handle the SIP sessions, regardless of which side of the FWLB deployment those sessions might originate from.
- When the communication session and SIP session are launched from different sides of an FWLB Deployment, the ACOS device can load balance sessions through the same firewall by using a persistent session template.

Topologies

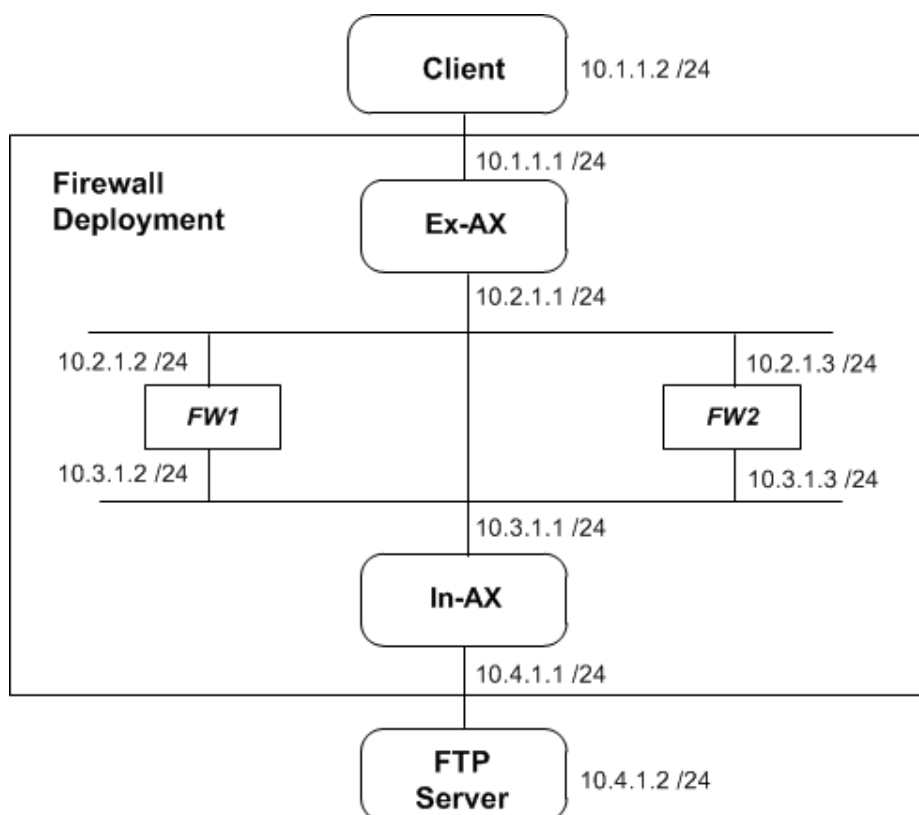
With a brief overview of FTP and SIP protocols out of the way, this next section provides a more in-depth illustration of sample network topologies. These topologies

provide the foundation for configuration examples that appear toward the end of this section.

FTP Network Diagram

[Figure 60](#) shows an example of a network topology for FTP FWLB.

Figure 60 : Topology for FTP FWLB



The network diagram has the following characteristics:

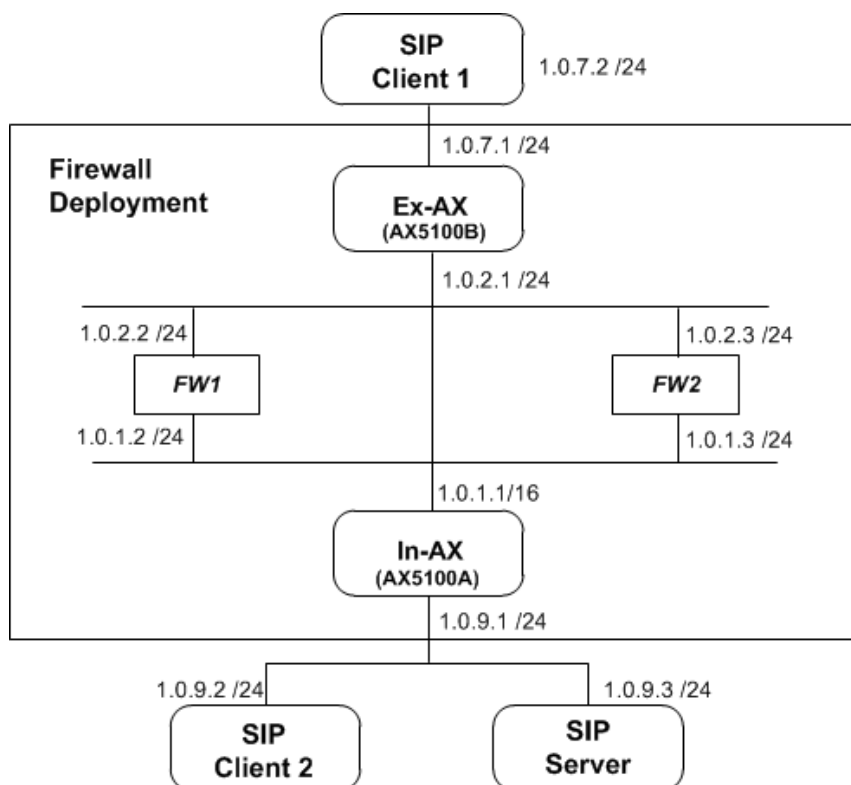
- A client is located at the top of a firewall deployment and an FTP server is located at the bottom. The firewalls are located at the center of the topology.
- The firewall deployment uses one external ACOS device ("Ex-AX") on the public side of the firewalls and an internal ACOS device ("In-AX") to handle traffic from the private side of the firewalls.

- Real-world scenarios often use two ACOS devices in VRRP-A high availability mode. However, for the sake of simplicity, the topology discussed in this section shows a single ACOS device on each side of the firewalls.
- Stateless load-balancing methods like Stateless Source IP Hash and Stateless Per-Packet Round Robin, are not supported for ALG protocols FWLB.

SIP Network Diagram

[Figure 61](#) shows another example of a network topology, but this illustration shows the network elements that would appear in a situation in which SIP traffic is being load balanced across a firewall deployment.

Figure 61 : Topology for SIP FWLB



The network diagram has the following characteristics:

- A SIP client is located at the top of the firewall deployment.
- A SIP client and a SIP server are located at the bottom of the firewall deployment.

- The firewall deployment uses one external ACOS device (“Ex-AX”) on the public side of the firewalls and an internal ACOS device (“In-AX”) to handle traffic from the private side of the firewalls.

Persistent Sessions

This section shows persistent session information for FTP and SIP. The persistent session information shown in the tables below correlates to the network topologies shown for FTP in [Topology for FTP FWLB](#), and for SIP in [Topology for SIP FWLB](#).

FTP Persistent Sessions

The two session tables below show persistent sessions for FTP FWLB.

- [Table 12](#) displays persistent sessions for the client-side session, from the perspective of the *external*-facing ACOS device.
- [Table 13](#) displays persistent sessions for the server-side session, from the perspective of the *internal*-facing ACOS device.

External-facing ACOS Device (client-side)

Session information in [Table 12](#) represents the control connection of an FTP transaction in passive FTP mode. The session (initiated from the client) is shown from the perspective of the external-facing device, “Ex-AX”.

Table 12 : ‘show session persist’ output for persistent FTP session through FWLB (“Ex-AX”)

Forward Source	Forward Dest	Reverse Source
10.1.1.2	10.4.1.2:65535	10.2.1.2:65535

- Forward Src – This leftmost column in the table above shows the IP address of the client (10.1.1.2). As with standard SLB scenarios, the Forward Source is the IP address of the client.
- Forward Dest – The Forward Destination, (middle column above), is the real server’s IP address (10.4.1.2). Note that this is different from standard SLB situations, in which the Forward Destination would usually be a VIP on the ACOS device instead of a real server.

- **Reverse Source** – The Reverse Source, (rightmost column above), represents the IP address (10.2.1.2) for the firewall interface connected to the external-facing ACOS device, “Ex-AX”. The fact that the firewall’s IP address is the Reverse Source is different from standard SLB scenarios where the Reverse Source would typically be the IP address of the VIP on the ACOS device.

Internal-facing ACOS Device (server-side)

The control connection for the server-side session, from the client to the server, opens a persistent session through the firewall. Subsequent TCP traffic, such as the data connection, has the same source IP and destination IP as the control connection, so it follows the same path and selects the same firewall as the persistent session selected by the control session.

[Table 13](#) below displays output from the show session persist command for the persistent sessions for passive FTP from the perspective of the internal-facing ACOS device (“In-AX”)

Table 13 : show session persist’ output for persistent FTP sessions through FWLB (“In-AX”)

Forward Source	Forward Dest	Reverse Source
10.1.1.2	10.4.1.2:65535	10.4.1.2:65535
10.4.1.2	10.1.1.2:65535	10.3.1.2:65535

The first session in the table is for the control session, and the second session is for the data session. The session output has the following noteworthy properties:

- **(Prot) Forward Src** – This is the IP address of the client (10.1.1.2). As with standard SLB scenarios, the Forward Source is also the IP address of the client.
- **Forward Dest** – The Forward Destination is the real server’s IP address (10.4.1.2). This is different from a standard SLB situation, in which the Forward Destination would usually be a VIP on the ACOS device.
- **Reverse Source** – This is the IP address of the real FTP server (10.4.1.2).
- The second session in the table can be disregarded because it belongs to the data connection; the data connection is following the control connection that was opened up by the first persistent session.

SIP Persistent Sessions

The two session tables below show persistent sessions for SIP firewall load balancing, as shown in the topology diagram in [Topology for SIP FWLB](#).

- [Table 14](#) displays persistent sessions for the server-side session, from the perspective of the *internal*-facing ACOS device.
- [Table 15](#) displays persistent sessions for the client-side session, from the perspective of the *external*-facing ACOS device.

Internal-facing ACOS Device (server-side)

The session tables below show persistent sessions for SIP FWLB. The server-side sessions are seen from the perspective of the internal-facing “In-AX” (AX5100A).

Table 14 : show session persist' output for persistent SIP session through FWLB (“In-AX”)

Forward Source	Forward Dest	Reverse Source
N/A	1.0.7.2:65535	1.0.1.2:65535

- Forward Src – This is a destination-IP persistent session. Therefore, there is no “Forward Source” port.
- Forward Dest – The Forward Destination, (middle column above), is the SIP client 1 in the public network (1.0.7.2). (See [SIP traffic originating from both sides of FWLB deployment](#).)
- Reverse Source – The Reverse Source, (rightmost column above), represents the IP address (1.0.1.2), which is the IP of the firewall interface connected to the internal-facing ACOS device, “ACOS5100A”.

External-facing ACOS Device (client-side)

The session information shown below represents the communication connection for a SIP transaction. The session (initiated from the client) is shown from the perspective of the external-facing device, “Ex-AX”.

When configuring SIP for FWLB, the source-IP persistence template and the destination-IP persistence template should be configured with the `netmask` option (with the value set to “/24”), in order to make the SIP server and SIP client 2 traffic

follow the same persistent session. The netmask option is needed because the two sessions have different IP addresses but are located within the same subnet.

Table 15 : 'show session persist' output for persistent SIP sessions through FWLB ("Ex-AX" - ACOS5100B)

Forward Source	Forward Dest	Reverse Source
1.0.7.2	1.0.9.3:65535	1.0.2.2:65535
1.0.7.2	1.0.9.2:65535	1.0.2.2:65535

The output for the two persistent sessions (from the perspective of the external-facing ACOS device, "ACOS5100B") has the following noteworthy properties:

- (Prot) Forward Src – The Forward Source is the IP address (1.0.7.2) associated with SIP client 1 on the public network.
- Forward Dest – The Forward Source is the IP address (1.0.9.3), which is associated with the SIP server in [Topology for SIP FWLB](#). (The second session in the table has an IP of 1.0.9.2, which is associated with SIP client 2.)
- Reverse Source – This address represents the IP (1.0.2.2) for the firewall interface connected to the external-facing ACOS device, "ACOS5100B".

Configuring ALG FWLB

This section shows how to configure an ACOS device for ALG FWLB using wildcard VIPs. Separate instructions are provided for FTP and SIP, and configuration instructions are provided for the ACOS GUI and CLI.

The process of configuring a pair of ACOS devices to handle AGL traffic, such as FTP and SIP, consists of the following high-level steps:

1. Create ACLs that are bound to the wildcard VIPs in order to permit traffic from specific clients or subnets. It is recommended that you use an extended ACL for greater granularity instead of a standard ACL.
2. Enable promiscuous mode on Ethernet interfaces connected to the firewalls, real servers, and clients.
3. Configure a Layer 3 ICMP health monitor with transparent mode enabled.

4. Configure the firewall nodes and real servers, and then add wildcard ports to the firewall nodes and disable the Layer 4 health checks on those ports.
5. Create separate service groups for the firewall nodes, real servers, and SIP or FTP clients.
6. Configure a source-IP persistence template and enable destination persistence.
7. Configure the wildcard VIPs:
 - Create a wildcard VIP for traffic in each direction.
 - Bind the ACL that matches traffic from the servers to the wildcard VIP for the servers.
 - Bind the ACL that matches traffic from the clients to the wildcard VIP for the firewalls.
 - Disable destination NAT.

Configuring ALG FWLB (CLI Procedure)

Configuring Use-rcv-hop-for-resp

The `use-rcv-hop-for-resp` command is used at the virtual port level to enable the ACOS device to support persistent sessions of ALG traffic across a firewall deployment.

Configuring Use-rcv-hop-group

To support firewall failover for long-lived sessions, configure the `use-rcv-hop-group` option. This feature enables FWLB to select an alternate firewall from the configured next hop server group if the firewall currently associated with an existing connection becomes unavailable.

NOTE: The `use-rcv-hop-group` feature applies only to traffic in the reverse direction (response traffic).

Additionally, to allow FWLB to automatically switch to another firewall when the active firewall handling connections becomes unavailable, enable the `re-select-if-server-down` option in the TCP template.

You can configure `use-rcv-hop-group` with the `ignore-mac` option. When `ignore-mac` is enabled, FWLB ignores MAC address information when selecting the next hop device from the server group. Instead, the next hop is selected based on the interface and VLAN associated with the traffic.

NOTE: ACOS supports a maximum of 500 server groups and up to 20 servers per server group. If either limit is exceeded, ACOS displays the message `Number of servers exceeds the current system or L3V partition limit.`

Configuring Src-dst-ip-swap-persist

The `use-rcv-hop-for-resp` command, with the `src-dst-ip-swap-persist` option, creates a persistent session after the source IP and destination IP are swapped. The new persistent session that is created should match both the source IP *and* the destination IP. This option should be used with the `incl-dst-ip` option.

This option cannot be used for the SIP protocol, because a SIP transaction may involve three or more parties.

Configuring Use-src-ip-for-dst-persist

The `use-rcv-hop-for-resp` command, with the `use-src-ip-for-dst-persist` option, at the virtual port level, creates a destination persistent session based on the source IP.

Configuring Use-dst-ip-for-src-persist

The `use-rcv-hop-for-resp` command, with the `use-dst-ip-for-src-persist` option, is used at the virtual port level. When this option is enabled, the ACOS device uses the destination IP to create source-IP persistent sessions for SIP or FTP sessions; the response packet goes through the same firewall as the client's request packet, and the SIP session and communication sessions is load balanced through the same firewall node.

Configuring Include destination IP on Persistence

The `incl-dst-ip` command is used within the source-IP persistence template for ALG protocols, such as FTP. A special persistent session is used for this feature, and this

option helps ensure that the session will be matched on both the source IP *and* destination IP addresses.

This option is not applicable to destination-IP persistence templates.

Configuring ALG FTP (CLI Example)

The CLI example below is based on the network topology for FTP FWLB shown in [Topology for FTP FWLB](#).

It is recommended that you use extended ACLs, rather than standard ACLs, in order to achieve greater granularity. Extended ACLs allow you to specify both the source and destination IP, so you have explicit control over which traffic is permitted and where it is allowed to go.

Configure ACLs (From Perspective of External ACOS Device)

This command creates extended “ACL 100”, which permits traffic from clients on subnet 10.1.1.0 and to FTP servers on subnet 10.4.1.0. (This ACL is later bound to the service group associated with the firewalls.)

```
ACOS(config)# access-list 100 permit tcp 10.1.1.0 0.0.0.255 10.4.1.0  
0.0.0.255
```

This command creates extended “ACL 101”, which permits traffic from FTP servers on subnet 10.4.1.0 and to clients on subnet 10.1.1.0. (This ACL is later bound to the service group associated with the clients.)

```
ACOS(config)# access-list 101 permit tcp 10.4.1.0 0.0.0.255 10.1.1.0  
0.0.0.255
```

Configure ACLs (From Perspective of Internal ACOS Device)

This command creates extended “ACL 150”, which permits traffic from clients on subnet 10.1.1.0 and to FTP servers on subnet 10.4.1.0. (This ACL is later bound to the service group associated with the FTP servers.)

```
ACOS(config)# access-list 150 permit tcp 10.1.1.0 0.0.0.255 10.4.1.0  
0.0.0.255
```

This command creates extended “ACL 151”, which permits traffic from FTP servers on subnet 10.4.1.0 and to clients on subnet 10.1.1.0. (This ACL is later bound to the service group associated with the firewalls.)

```
ACOS(config)# access-list 151 permit tcp 10.4.1.0 0.0.255.255 10.1.1.0 0.0.255.255
```

Configure Ethernet Interfaces and Enable Promiscuity

The following commands create the Ethernet interfaces connected to the firewalls and the real servers or clients, and then enable promiscuous mode:

```
ACOS(config)# interface ethernet 1  
ACOS(config-if:ethernet:1)# ip address 10.3.1.1 255.255.255.0  
ACOS(config-if:ethernet:1)# ip allow-promiscuous-vip  
ACOS(config-if:ethernet:1)# exit  
ACOS(config)# interface ethernet 2  
ACOS(config-if:ethernet:2)# ip address 10.4.1.1 255.255.255.0  
ACOS(config-if:ethernet:2)# ip allow-promiscuous-vip  
ACOS(config-if:ethernet:2)# exit
```

Health Monitor for Internal ACOS device

These commands create health monitor “FW-HC”, which contains the method ICMP transparent. The method performs a transparent health check and sends a ping through the firewall to the IP address of the external-facing ACOS device (“Ex-AX”) at 10.2.1.1.

```
ACOS(config)# health monitor FW-HC  
ACOS(config-health:monitor)# method icmp transparent 10.2.1.1  
ACOS(config-health:monitor)# exit
```

Configure Firewall Nodes and Real Server

Next, create a server configuration for the firewall node “FW1” (at IP address 10.3.1.2) and another firewall node “FW2” (at IP address 10.3.1.3), and assign the “FW-HC” health check. Then, add wildcard ports (port 0) to the firewall nodes.

Create a server configuration for the FTP server, “SRV”, at IP address 10.4.1.2, and add wildcard ports (port 0) to the server while disabling the Layer 4 health checks, which are enabled by default.

While it may seem unusual to do so, create a server configuration for the client, “CL” (at IP address 10.1.1.2). This ensures the FTP sessions are correctly routed across the firewall while maintaining session persistence. As with the other servers, you must add wildcard ports (port 0) while disabling the Layer 4 health checks.

```
ACOS(config)# slb server FW1 10.3.1.2
ACOS(config-real server)# health-check FW-HC
ACOS(config-real server)# port 0 tcp
ACOS(config-real server-node port)# health-check-disable
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
ACOS(config)# slb server FW2 10.3.1.3
ACOS(config-real server)# health-check FW-HC
ACOS(config-real server)# port 0 tcp
ACOS(config-real server-node port)# health-check-disable
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
ACOS(config)# slb server SRV 10.4.1.2
ACOS(config-real server)# health-check-disable
ACOS(config-real server)# port 0 tcp
ACOS(config-real server-node port)# health-check-disable
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
ACOS(config)# slb server CL 10.1.1.2
ACOS(config-real server)# health-check-disable
ACOS(config-real server)# port 0 tcp
ACOS(config-real server-node port)# health-check-disable
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
```

Create Service Groups for Firewalls, Real Server, and Clients

These commands create the service group, “FW-SG”, which will be used to manage the two firewall nodes. Then, add the two firewall nodes, “FW1” and “FW2”, as service group members, on port 0. Similarly, create a separate service group “SRV-SG” to manage the FTP server, and then add the server “SRV” on port 0. Lastly, create a separate service group “CL-SG” to manage the clients, and then add the client “CL” on port 0.

```
ACOS(config)# slb service-group FW-SG tcp
ACOS(config-slb svc group)# member FW1 0
```

```

ACOS(config-slb svc group-member:0)# exit
ACOS(config-slb svc group)# member FW2 0
ACOS(config-slb svc group-member:0)# exit
ACOS(config-slb svc group)# exit
ACOS(config)# slb service-group SRV-SG tcp
ACOS(config-slb svc group)# member SRV 0
ACOS(config-slb svc group-member:0)# exit
ACOS(config-slb svc group)# exit
ACOS(config)# slb service-group CL-SG tcp
ACOS(config-slb svc group)# member CL 0
ACOS(config-slb svc group-member:0)# exit
ACOS(config-slb svc group)# exit

```

Configure a Source IP Persistence Template

Create a source-IP persistence template “AAA” and use the `incl-dst-ip` command to enable destination persistence.

```

ACOS(config)# slb template persist source-ip aaa
ACOS(config-source ip persist)# incl-dst-ip
ACOS(config-source ip persist)# exit

```

Configure Wildcard VIPs

Use the following commands to create the wildcard VIPs at IP address 0.0.0.0, which will be used to handle FTP traffic coming to the external-facing ACOS device from the client on the public network. The previously-created ACL (“ACL 100”) is bound to the wildcard VIP, port 0 is associated with service group “FW-SG”, destination NAT is disabled, and persistence is enabled.

```

ACOS(config)# slb virtual-server toFW 0.0.0.0 acl 100
ACOS(config-slb vserver)# port 0 tcp
ACOS(config-slb vserver-vport)# name _wildcard_v4_100_TCP_65535
ACOS(config-slb vserver-vport)# service-group FW-SG
ACOS(config-slb vserver-vport)# no-dest-nat
ACOS(config-slb vserver-vport)# template persist source-ip aaa
ACOS(config-slb vserver-vport)# exit
ACOS(config-slb vserver)# exit

```

These commands create the wildcard VIPs at IP address 0.0.0.0 to handle FTP traffic *from* the external-facing ACOS device to real servers on the private network. “ACL

101” is bound to the wildcard VIP; port 0 is associated with service group “SRV-SG”; destination NAT is disabled; and persistence is enabled.

```
ACOS(config)# slb virtual-server fromFW 0.0.0.0 acl 101
ACOS(config-slb vserver)# port 0 tcp
ACOS(config-slb vserver-vport)# name _wildcard_v4_101_TCP_65535
ACOS(config-slb vserver-vport)# service-group SRV-SG
ACOS(config-slb vserver-vport)# use-rcv-hop-for-resp src-dst-ip-swap-
persist
ACOS(config-slb vserver-vport)# no-dest-nat
ACOS(config-slb vserver-vport)# template persist source-ip aaa
ACOS(config-slb vserver-vport)# exit
ACOS(config-slb vserver)# exit
```

Verifying FTP Configurations with Show Command

When finished with these configurations, you can use the “show session” command to verify that an FTP control connection could create a src-dst-ip-swap-persist session.

```
Internal-ACOS(config)# show session
```

Port	Forward Source	Forward Dest	Reverse Source
src 10.4.1.2		10.1.1.2:65535	10.3.1.3:65535
Total Sessions:		1	

When FTP control connection is established, data connections select the right firewall using the established persistent session.

Configuring ALG SIP – Internal Device (CLI Example)

The configurations below are based on the network topology shown in [Topology for SIP FWLB](#) and represent the configuration that must be made on the *internal*-facing ACOS device (ACOS5100A).

Configure ACLs

The following commands create a standard ACL, which will be applied to the internal-facing ACOS device (“ACOS5100A”), and will permit traffic from “SIP client 1”, which is

located on the public subnet (1.0.7.0). At the same time, this ACL will deny all other traffic.

```
Internal-ACOS(config)# access-list 2 10 permit 1.0.7.0 0.0.0.255 log
```

The following commands create standard ACL, which will be applied to the internal-facing ACOS device ("ACOS5100A"), and will permit traffic from the SIP client and SIP server on the internal subnet (1.0.9.0) while denying all other traffic.

```
Internal-ACOS(config)# access-list 3 10 permit 1.0.9.0 0.0.0.255 log
```

Configure Ethernet Interfaces and Enable Promiscuity

The following commands create the Ethernet interfaces on the *internal*-facing ACOS device that is connected to the firewalls and to the real servers or clients. Then, commands are used to enable promiscuous mode on those interfaces:

```
Internal-ACOS(config)# interface ethernet 1  
Internal-ACOS(config-if:ethernet:1)# ip address 1.0.9.1 255.255.255.0  
Internal-ACOS(config-if:ethernet:1)# ip allow-promiscuous-vip  
Internal-ACOS(config-if:ethernet:1)# exit  
Internal-ACOS(config)# interface ethernet 2  
Internal-ACOS(config-if:ethernet:1)# ip address 1.0.1.1 255.255.255.0  
Internal-ACOS(config-if:ethernet:1)# ip allow-promiscuous-vip
```

Configure Health Monitor on Internal ACOS device

The following command creates the health monitor "fw-hc1", which contains the method ICMP transparent. The method is used to perform a transparent health check, and it will send a ping through the firewall to the ACOS interface on the far side of the firewall at IP 1.0.2.1:

```
Internal-ACOS(config)# health monitor fw-hc1  
Internal-ACOS(config-health:monitor)# method icmp transparent 1.0.2.1
```

Configure SIP Clients, Firewall Nodes, and SIP Servers

Next, create a server configuration for the SIP client, "sip-client2", which is at IP address 1.0.9.2. This is necessary to ensure the SIP sessions can be correctly routed across the firewall while maintaining session persistence. Add wildcard ports at port

0 for both TCP and UDP, both of which are necessary for SIP. In addition, disable Layer 4 health checks on both wildcard ports.

```
Internal-ACOS(config)# slb server sip-client2 1.0.9.2
Internal-ACOS(config-real server)# port 0 tcp
Internal-ACOS(config-real server-node port)# health-check-disable
Internal-ACOS(config-real server-node port)# exit
Internal-ACOS(config-real server)# port 0 udp
Internal-ACOS(config-real server-node port)# health-check-disable
Internal-ACOS(config-real server-node port)# exit
Internal-ACOS(config-real server)# exit
```

Create a server configuration for the firewall node “fw1” (at IP address 1.0.1.2) and firewall node “fw2” (at IP address 1.0.1.3), and assign the “fw-hc1” health check to both firewall nodes. Add wildcard ports (port 0) to the firewall nodes for TCP and UDP, and disable the Layer 4 health checks for these wildcard ports.

```
Internal-ACOS(config)# slb server fw1 1.0.1.2
Internal-ACOS(config-real server)# health-check fw-hc1
Internal-ACOS(config-real server)# port 0 tcp
Internal-ACOS(config-real server-node port)# health-check-disable
Internal-ACOS(config-real server-node port)# exit
Internal-ACOS(config-real server)# port 0 udp
Internal-ACOS(config-real server-node port)# health-check-disable
Internal-ACOS(config-real server-node port)# exit
Internal-ACOS(config-real server)# exit
Internal-ACOS(config)# slb server fw2 1.0.1.3
Internal-ACOS(config-real server)# health-check fw-hc1
Internal-ACOS(config-real server)# port 0 tcp
Internal-ACOS(config-real server-node port)# health-check-disable
Internal-ACOS(config-real server-node port)# exit
Internal-ACOS(config-real server)# port 0 udp
Internal-ACOS(config-real server-node port)# health-check-disable
Internal-ACOS(config-real server-node port)# exit
Internal-ACOS(config-real server)# exit
```

Next, create a server configuration for the SIP server “sip-srv” (at IP address 1.0.9.3), and add wildcard ports (port 0) for TCP and UDP, while disabling the Layer 4 health checks on port 0, which are enabled by default.

```
Internal-ACOS(config)# slb server sip-srv 1.0.9.3
Internal-ACOS(config-real server)# health-check-disable
```

```
Internal-ACOS(config-real server)# port 0 tcp
Internal-ACOS(config-real server-node port)# health-check-disable
Internal-ACOS(config-real server-node port)# exit
Internal-ACOS(config-real server)# port 0 udp
Internal-ACOS(config-real server-node port)# health-check-disable
Internal-ACOS(config-real server-node port)# exit
Internal-ACOS(config-real server)# exit
```

Create the Service Groups for Clients, Firewalls, and Server

Create two service group “sg-sip-client2-tcp” and “sg-sip-client2-udp” to manage the clients, and then add the client “sip-client2” to each service group at port 0.

```
Internal-ACOS(config)# slb service-group sg-sip-client2-tcp tcp
Internal-ACOS(config-slb svc group)# member sip-client2 0
Internal-ACOS(config-slb svc group-member:0)# exit
Internal-ACOS(config-slb svc group)# exit
Internal-ACOS(config)# slb service-group sg-sip-client2-udp udp
Internal-ACOS(config-slb svc group)# member sip-client2 0
Internal-ACOS(config-slb svc group-member:0)# exit
Internal-ACOS(config-slb svc group)# exit
```

Use these commands to create service groups “sg-fw-tcp1” (TCP) and “sg-fw-udp2” (UDP). These service groups manage the firewall nodes. Add two firewall nodes, “fw1” and “fw2”, on port 0 to each service group.

```
Internal-ACOS(config)# slb service-group sg-fw-tcp1 tcp
Internal-ACOS(config-slb svc group)# member fw1 0
Internal-ACOS(config-slb svc group-member:0)# exit
Internal-ACOS(config-slb svc group)# member fw2 0
Internal-ACOS(config-slb svc group-member:0)# exit
Internal-ACOS(config-slb svc group)# exit
Internal-ACOS(config)# slb service-group sg-fw-udp2 udp
Internal-ACOS(config-slb svc group)# member fw1 0
Internal-ACOS(config-slb svc group-member:0)# exit
Internal-ACOS(config-slb svc group)# member fw2 0
Internal-ACOS(config-slb svc group-member:0)# exit
Internal-ACOS(config-slb svc group)# exit
```


Similarly, create two separate service groups to manage the SIP server. Create service group “sg-sip-srv1” for TCP traffic and create “sg-sip-srv2”. Then, add “sip-srv” as a member of each service group on port 0.

```
Internal-ACOS(config)# slb service-group sg-sip-srv1 tcp
Internal-ACOS(config-slb svc group)# member sip-srv 0
Internal-ACOS(config-slb svc group-member:0)# exit
Internal-ACOS(config-slb svc group)# exit
Internal-ACOS(config)# slb service-group sg-sip-srv2 udp
Internal-ACOS(config-slb svc group)# member sip-srv 0
Internal-ACOS(config-slb svc group-member:0)# exit
Internal-ACOS(config-slb svc group)# exit
```

Configure a Destination IP Persistence Template

Create a destination-IP persistence template “dtemp1”:

```
Internal-ACOS(config)# slb template persist destination-ip dtemp1
Internal-ACOS(config-destination ip persist)# exit
```

Configure Wildcard VIPs on the Internal-facing ACOS Device

Use the following commands to create the wildcard VIPs at IP address 0.0.0.0. This wildcard VIP will be assigned to the internal-facing ACOS device (ACOS5100A), and will be used to handle SIP traffic coming *from* the firewall (and from the SIP client on the public network), and going to the internal-facing ACOS device.

The previously-created ACL is bound to the wildcard VIP, thus allowing traffic from SIP client 1, while denying all other traffic.

Port 0 is associated with service groups “sg-sip-client2-tcp” and “sg-sip-client2-udp”. Destination NAT is disabled.

The use-rcv-hop-for-resp use-src-ip-for-dst-persist option is applied to wildcard ports for both TCP and UDP to create a destination persistent session based on source IP. The no-dest-nat option is applied to TCP and UDP service groups to help ensure the session is matched on both the source IP *and* destination IP addresses.

```
Internal-ACOS(config)# slb virtual-server fromFW 0.0.0.0 acl 2
Internal-ACOS(config-slb vserver)# port 0 tcp
Internal-ACOS(config-slb vserver-vport)# service-group sg-sip-client2-tcp
```

```
Internal-ACOS(config-slb vserver-vport)# use-rcv-hop-for-resp use-src-ip-  
for-dst-persist  
Internal-ACOS(config-slb vserver-vport)# no-dest-nat  
Internal-ACOS(config-slb vserver-vport)# exit  
Internal-ACOS(config-slb vserver)# port 0 udp  
Internal-ACOS(config-slb vserver-vport)# service-group sg-sip-client2-udp  
Internal-ACOS(config-slb vserver-vport)# use-rcv-hop-for-resp use-src-ip-  
for-dst-persist  
Internal-ACOS(config-slb vserver-vport)# no-dest-nat  
Internal-ACOS(config-slb vserver-vport)# exit  
Internal-ACOS(config-slb vserver)# exit
```

Use these commands to create the wildcard VIPs at IP address 0.0.0.0. This wildcard VIP is assigned to the internal-facing ACOS device (ACOS5100A) and used to handle SIP traffic going to the firewall from the internal-facing ACOS device (and originally from the SIP client and SIP server on the private/internal network).

The previously-created ACL is bound to the wildcard VIP, thus allowing traffic from internal subnet (that is, SIP client 2 and the SIP server), while denying all other traffic. In addition, port 0 is associated with service group “sg-fw-tcp1” and “sg-fw-udp2”. Destination NAT is disabled.

Also, the no-dest-nat option is applied to the TCP and UDP service groups to help ensure the session will be matched on both the source IP *and* destination IP addresses.

```
Internal-ACOS(config)# slb virtual-server toFW 0.0.0.0 acl 3  
Internal-ACOS(config-slb vserver)# port 0 tcp  
Internal-ACOS(config-slb vserver-vport)# service-group sg-fw-tcp1  
Internal-ACOS(config-slb vserver-vport)# no-dest-nat  
Internal-ACOS(config-slb vserver-vport)# template persist destination-ip  
dtempl  
Internal-ACOS(config-slb vserver-vport)# exit  
Internal-ACOS(config-slb vserver)# port 0 udp  
Internal-ACOS(config-slb vserver-vport)# service-group sg-fw-udp2  
Internal-ACOS(config-slb vserver-vport)# no-dest-nat  
Internal-ACOS(config-slb vserver-vport)# template persist destination-ip  
dtempl  
Internal-ACOS(config-slb vserver-vport)# exit  
Internal-ACOS(config-slb vserver)# exit
```

Verifying Internal-ACOS SIP Configurations with Show Command

When finished with these configurations, you can use the “show session” command to verify that a SIP control connection could create a dst-ip-persistent session, as shown below:

```
Internal-ACOS(config)# show session
```

Prot	Forward Dest	Reverse Source	Age
dst	1.0.7.2:65535	1.0.1.2:65535	300
Total Sessions:		1	

Configuring ALG SIP – External Device (CLI Example)

The configurations below are based on the network topology shown in [Topology for SIP FWLB](#) and represent the configurations that must be made on the *external*-facing ACOS device (ACOS5100B).

Configure ACLs

This command creates a standard ACL, which is applied to the external-facing ACOS device (“ACOS5100B”), and permits traffic from “SIP client 2” located on the private subnet (1.0.9.0). This ACL denies all other traffic.

```
External-ACOS(config)# access-list 4 10 permit 1.0.9.0 0.0.0.255 log
```

This command creates a standard ACL, which is applied to the external-facing ACOS device (“ACOS5100B”), and permits traffic from SIP client 1 on the public subnet (1.0.7.0) while denying all other traffic.

```
External-ACOS(config)# access-list 5 10 permit 1.0.7.0 0.0.0.255 log
```

Configure Ethernet Interfaces and Enable Promiscuity

The following commands create the Ethernet interfaces on the *external*-facing ACOS device that is connected to the firewalls and to the SIP client. Promiscuous mode is then enabled on those same interfaces:

```
External-ACOS(config)# interface ethernet 1
External-ACOS(config-if:ethernet:1)# ip address 1.0.2.1 255.255.255.0
External-ACOS(config-if:ethernet:1)# ip allow-promiscuous-vip
External-ACOS(config-if:ethernet:1)# exit
```

```
External-ACOS(config)# interface ethernet 2  
External-ACOS(config-if:ethernet:1)# ip address 1.0.7.1 255.255.255.0  
External-ACOS(config-if:ethernet:1)# ip allow-promiscuous-vip  
External-ACOS(config-if:ethernet:1)# exit
```

Configure health monitor on External ACOS device

The following command creates the health monitor “fw-hc2”, which contains the method ICMP transparent. The method is used to perform a transparent health check by sending a ping through the firewall to the ACOS interface on the far side of the firewall at IP 1.0.1.1:

```
External-ACOS(config)# health monitor fw-hc2  
External-ACOS(config-health:monitor)# method icmp transparent 1.0.1.1  
External-ACOS(config-health:monitor)# exit
```

Configure SIP Clients, Firewall Nodes, and SIP Servers

A server configuration is created for the SIP client, “sip-client1” (at IP 1.0.7.2). This is necessary to ensure the SIP sessions can be correctly routed across the firewall while maintaining session persistence. Add wildcard ports (port 0) for both TCP and UDP, both of which are necessary for SIP. In addition, disable the Layer 4 health checks these wildcard ports.

```
External-ACOS(config)# slb server sip-client1 1.0.7.2  
External-ACOS(config-real server)# port 0 tcp  
External-ACOS(config-real server-node port)# exit  
External-ACOS(config-real server)# port 0 udp  
External-ACOS(config-real server-node port)# exit  
External-ACOS(config-real server)# exit
```

Create a server configuration for the firewall node “fw1” (at IP 1.0.2.2) and firewall node “fw2” (at IP 1.0.2.3). Assign the health check created earlier (“fw-hc2”). Then, add wildcard ports (port 0) to the firewall nodes for TCP and UDP, while disabling the default layer 4 health checks.

```
External-ACOS(config)# slb server fw1 1.0.2.2  
External-ACOS(config-real server)# health-check fw-hc2  
External-ACOS(config-real server)# port 0 tcp  
External-ACOS(config-real server-node port)# health-check-disable  
External-ACOS(config-real server-node port)# exit
```

```

External-ACOS(config-real server)# port 0 udp
External-ACOS(config-real server-node port)# health-check-disable
External-ACOS(config-real server-node port)# exit
External-ACOS(config-real server)# exit
External-ACOS(config)# slb server fw2 1.0.2.3
External-ACOS(config-real server)# health-check fw-hc2
External-ACOS(config-real server)# port 0 tcp
External-ACOS(config-real server-node port)# health-check-disable
External-ACOS(config-real server-node port)# exit
External-ACOS(config-real server)# port 0 udp
External-ACOS(config-real server-node port)# health-check-disable
External-ACOS(config-real server-node port)# exit
External-ACOS(config-real server)# exit

```

There is no server configuration required for the SIP server because that device is connected to the ACOS5100A, the *internal*-facing ACOS device.

Create the Service Groups for Clients, Firewalls, and Server

Create a service group, “sg-sip-client1-tcp” to manage the TCP portion of the SIP transaction on the client, and then add “sip-client1” on port 0 of the service group.

```

External-ACOS(config)# slb service-group sg-sip-client1-tcp tcp
External-ACOS(config-slbg svc group)# member sip-client1 0
External-ACOS(config-slbg svc group-member:0)# exit
External-ACOS(config-slbg svc group)# exit

```

Repeat the process above for UDP. Create a service group, “sg-sip-client1-udp” to manage the UDP portion of the SIP transaction on the client, and then add “sip-client1” on port 0 of the service group.

```

External-ACOS(config)# slb service-group sg-sip-client1-udp udp
External-ACOS(config-slbg svc group)# member sip-client1 0
External-ACOS(config-slbg svc group-member:0)# exit
External-ACOS(config-slbg svc group)# exit

```

These commands create service groups “sg-fw-tcp3” (TCP) and “sg-fw-udp4” (UDP), which are used to manage firewall nodes. The firewall nodes, members “fw1” and “fw2”, are added to port 0 of each service group.

```

External-ACOS(config)# slb service-group sg-fw-tcp3 tcp
External-ACOS(config-slbg svc group)# member fw1 0
External-ACOS(config-slbg svc group-member:0)# exit

```

```
External-ACOS(config-slb svc group)# member fw2 0
External-ACOS(config-slb svc group-member:0)# exit
External-ACOS(config-slb svc group)# exit
External-ACOS(config)# slb service-group sg-fw-udp4 udp
External-ACOS(config-slb svc group)# member fw1 0
External-ACOS(config-slb svc group-member:0)# exit
External-ACOS(config-slb svc group)# member fw2 0
External-ACOS(config-slb svc group-member:0)# exit
External-ACOS(config-slb svc group)# exit
```

There is no service group required for the SIP server because that device is connected to the *internal*-facing ACOS device (ACOS5100A).

Configure a Destination IP Persistence Template

Use these commands to create a source-IP persistence template “stemp1” with incl-dst-ip option enabled:

```
External-ACOS(config)# slb template persist source-ip stemp1
External-ACOS(config-source ip persist)# incl-dst-ip
External-ACOS(config-source ip persist)# exit
```

Configure Wildcard VIPs on the External-facing ACOS Device

Use these commands to create the wildcard VIPs at IP address 0.0.0.0. This wildcard VIP will be assigned to the external-facing ACOS device (ACOS5100B), and will be used to handle SIP traffic coming *from* the firewall, (and from the SIP client on the private network), and going *to* the external-facing ACOS device.

The previously-created ACL is applied to the wildcard VIP, thus allowing traffic from SIP client 2, while denying all other traffic.

In addition, port 0 is associated with service group “sg-sip-client1-tcp” and “sg-sip-client1-udp”, and destination NAT is disabled. Also, the use-rcv-hop-for-resp use-dst-ip-for-src-persist command is applied to the wildcard ports (port 0), which will cause the response packets to go through the same firewall as the client’s request packets. The communication and SIP sessions will be load balanced through the same firewall node.

```
External-ACOS(config)# slb virtual-server fromFW 0.0.0.0 acl 4
External-ACOS(config-slb vserver)# port 0 tcp
External-ACOS(config-slb vserver-vport)# service-group sg-sip-client1-tcp
```

```

External-ACOS(config-slb vserver-vport) # use-rcv-hop-for-resp use-dst-ip-
for-src-persist
External-ACOS(config-slb vserver-vport) # no-dest-nat
External-ACOS(config-slb vserver-vport) # exit
External-ACOS(config-slb vserver) # port 0 udp
External-ACOS(config-slb vserver-vport) # service-group sg-sip-client1-udp
External-ACOS(config-slb vserver-vport) # use-rcv-hop-for-resp use-dst-ip-
for-src-persist
External-ACOS(config-slb vserver-vport) # no-dest-nat
External-ACOS(config-slb vserver-vport) # exit
External-ACOS(config-slb vserver) # exit

```

Use the following commands to create the wildcard VIPs at IP address 0.0.0.0. This wildcard VIP will be assigned to the external-facing ACOS device (ACOS5100B), and will be used to handle SIP traffic coming to the external-facing ACOS device from the SIP client. The previously-created ACL is bound to the wildcard VIP, thus allowing traffic from public network (that is, SIP client 1), while denying all other traffic. In addition, port 0 is associated with service group “sg-fw-tcp” and “sg-fw-udp”. Destination NAT is disabled.

```

External-ACOS(config) # slb virtual-server toFW 0.0.0.0 acl 5
External-ACOS(config-slb vserver) # port 0 tcp
External-ACOS(config-slb vserver-vport) # service-group sg-fw-tcp3
External-ACOS(config-slb vserver-vport) # no-dest-nat
External-ACOS(config-slb vserver-vport) # template persist source-ip stemp1
External-ACOS(config-slb vserver-vport) # exit
External-ACOS(config-slb vserver) # port 0 udp
External-ACOS(config-slb vserver-vport) # service-group sg-fw-udp4
External-ACOS(config-slb vserver-vport) # no-dest-nat
External-ACOS(config-slb vserver-vport) # template persist source-ip stemp1
External-ACOS(config-slb vserver-vport) # exit
External-ACOS(config-slb vserver) # exit

```

Verifying ACOS5100B SIP Configurations with Show Command

When finished with these configurations, you can use the “show session” command to see that a SIP session could create the following source-IP persistent sessions, as shown below:

```

Internal-ACOS(config) # show session

```

Prot	Forward Source	Forward Dest	Reverse Source
Age			

src 1.0.7.2		1.0.9.3:65535	1.0.2.2:65535
300			
src 1.0.7.2		1.0.9.2:65535	1.0.2.2:65535
300			
Internal-ACOS(config)#			

NOTE: The two sessions in the table are the same. This is because the SIP session is following the persistent session that has already been established by the communication session.

Wildcard VIP

A wildcard VIP is a VIP that does not have a specific IP address. Instead, wildcard VIPs have IP address 0.0.0.0 (for IPv4) or “::” (for IPv6). Client requests sent to any IP address will be accepted when they are received at a wildcard VIP.

Wildcard VIPs use Access Control Lists (ACLs) to filter client requests, thus preventing potential miscreants from causing damage to network resources located behind the wildcard VIP.

To load balance ALG protocols through the firewalls, wildcard VIPs are necessary to handle traffic heading in each direction (ingress and egress). A pair of wildcard VIPs is needed for each ACOS device. One wildcard VIP is needed for traffic coming *to* the firewall and another is needed to handle traffic going *from* the firewall.

ACLs for Identifying Valid Traffic

To specify the source and destination IP addresses that a wildcard VIP will accept from clients, a pair of ACLs must be configured. Extended ACLs, which can filter based on destination address, IP protocol, and TCP/UDP port numbers, should be used to provide more granular control. The goal is to permit traffic along the path from a specific client subnet to the IP addresses of the real servers.

- Each ACL has an implicit “deny any” rule at the end. Any traffic that is not explicitly permitted by another rule in the ACL is denied by the implicit “deny any” rule.
- ACOS can have only one wildcard VIP that is not bound to an ACL. This unbound wildcard VIP is known as the default, and it is used to process traffic that does not create a positive match on any of the other ACLs that are bound to other wildcard VIPs.

From External ACOS Perspective (Client-Side Session):

- An ACL is configured to permit traffic from the client subnet (10.1.1.x/24) to pass to the server’s subnet (10.4.1.x/24). This ACL is bound to the service group associated with the firewalls. (In the sample configuration that follows, this ACL is called “ACL 100”.)
- An ACL is configured to permit traffic from the server subnet (10.4.1.x/24) to pass to the client’s subnet (10.1.1.x/24). This ACL is bound to the service group associated with the client. (In the sample configuration that follows, this ACL is called “ACL 101”.)

From Internal ACOS Perspective (Server-Side Session):

- An ACL is configured to permit traffic from the client subnet (10.1.1.x/24) to pass to the server’s subnet (10.4.1.x/24). This ACL is bound to the service group associated with the servers. (In the sample configuration that follows, this ACL is called “ACL 150”.)
- An ACL is configured to permit traffic from the server subnet (10.4.1.x/24) to pass to the client’s subnet (10.1.1.x/24). This ACL is bound to the service group associated with the firewalls. (In the sample configuration that follows, this ACL is called “ACL 151”.)

Wildcard Protocol Ports on the Wildcard

Each VIP on the ACOS devices in an ALG protocol FWLB deployment (“Ex-AX” and “In-AX”) requires a wildcard TCP port (port 0). This wildcard port 0 is required because the data session destination port is unknown when dealing with ALG scenarios. Thus, simply configuring well-known FTP ports 20 and 21 will not work and a wildcard port must be used instead.

These parameters (enabled by default) must be disabled for ALG protocol FWLB to work:

- Layer 4 health check – Layer 4 health checks are typically used to test the TCP ports on a real server. The health check consists of a TCP SYN request being sent to the TCP port on an ACOS device. If the TCP port responds with a TCP SYN ACK, then it is determined that the TCP port is healthy. If no response is received, then it is determined that the TCP port is down.

The Layer 4 health check must be disabled in ALG protocol FWLB scenarios. If not disabled, then the default Layer 4 health check method will be used, causing the SYN packet to be sent to the default port (“port 65535”) of the real server. The SYN packet will *not* be received on port 0, the SYN ACK response will not be sent, and the health check will fail (because it will appear as though the real server status is down).

- Destination NAT – With destination NAT enabled, the ACOS device swaps the destination address in the packet from the client with the VIP on the ACOS device. However, destination NAT must be disabled at the wildcard port level to prevent this swap or else the traffic from the client will have its destination IP changed to the IP address of the service-group member. This would be the IP address of the firewall, which would present problems because the firewall can not handle the traffic.

Server and Service-group Requirements

- On the external-facing ACOS device (“Ex-AX”), a service group is required for the firewalls, and another service group is required for the client. However, in most deployments, the service group would be configured for the next-hop router(s) instead of an actual client.
- On the internal ACOS device (“In-AX”), a service group is required for the firewalls, and another service group is required for the real server(s).
- When configuring the service groups, keep in mind that stateless load balancing algorithms, such as Stateless Source IP Hash, are not supported.
- All real servers in the service groups must use wildcard ports, such as “port 0 tcp”. If this is not done, persistent FWLB for FTP will not work correctly. The FTP protocol uses well-known ports 20 and 21, so specifying just one of these ports would result in traffic from the other port being denied.
- Layer 4 health checks on the wildcard ports must be disabled. If this is not done, the configuration will fail because the Layer 4 health check traffic will be sent to

default port 65535 instead of port 0. No response will be generated, and it will appear as though the port is down.

Promiscuous Mode

Promiscuous mode can be enabled on ACOS data interfaces. By default, the option is disabled, but it must be enabled on the data interfaces that are connected to the firewalls in an ALG FWLB configuration.

When using a Virtual Ethernet (VE) interface to connect to the firewalls, you must enable promiscuous mode only on the VE itself. ACOS automatically enables promiscuous mode on each of the Ethernet ports in the VLAN that belongs to that VE.

Session Persistence for FTP and SIP

Load balancing ALG protocols across a firewall requires session persistence. For a given session, the same firewall must be used for traffic in both directions. For example, if the ACOS device selects FW1 (see [Topology for FTP FWLB](#)) for a client's FTP request, the ACOS device must continue to use FW1 for all subsequent traffic on that control session.

FTP-Specific Configurations

FWLB for FTP requires the following configuration options for persistence:

- Source-IP persistence template – Within the source-IP persistence template, the include destination-IP option is used to enable persistence of the destination IP address. When the include destination-IP option is enabled in a source-IP persistence template, the ACOS device uses the same firewall for a given source-destination pair of IP addresses for the entire session. This option is disabled by default.
- The internal ACOS device ("In-AX"), which is connected to the FTP servers, must have the persistence template configured on *both* wildcard VIPs. However, the external ACOS device ("Ex-AX"), which is connected to the clients, only needs to have the persistence template configured on the wildcard VIP that is bound to the firewalls, but not on the wildcard VIP that is bound to the client.

- Use-rcv-hop-for-resp – This option is configurable on individual virtual ports. When enabled, it forces the ACOS device to send a reply back through the last hop from which the request was received. The use-rcv-hop-for-resp option has several sub-options that can be used with the SIP protocol that can use more than two sessions.
 - On the ACOS device connected to clients, FWLB ALG for FTP requires this option on the wildcard virtual port for the server-to-client direction. This is the virtual port on the wildcard VIP that uses that ACL that matches on the client subnet.
 - On the ACOS device connected to the servers, the use-rcv-hop-for-resp option is required on the wildcard virtual port for the client-to-server direction. In this case, the src-dst-ip-swap-persist suboption also is required. The src-dst-ip-swap-persist suboption “swaps” the source and destination addresses in the persistent session.

SIP-Specific Configuration

FWLB for SIP requires the following configuration options for persistence:

- Destination-IP session persistence should be configured on the ACOS device that is connected to the SIP server and SIP client 2. (See [Topology for SIP FWLB.](#))
- Source-IP session persistence should be configured (using the incl-dst-ip command) on the ACOS device that is connected to SIP client 1. (See [Topology for SIP FWLB.](#))

In order to get both sessions (originating from different sides of the FWLB) to go through the same firewall node, a special persistent session must be configured on the ACOS device at the side of the SIP server and SIP client 2. (See [Topology for SIP FWLB.](#))

The options: use-src-ip-for-dst-persist and use-dst-ip-for-src-persist, have the same operation mode as src-dst-ip-swap-persist.

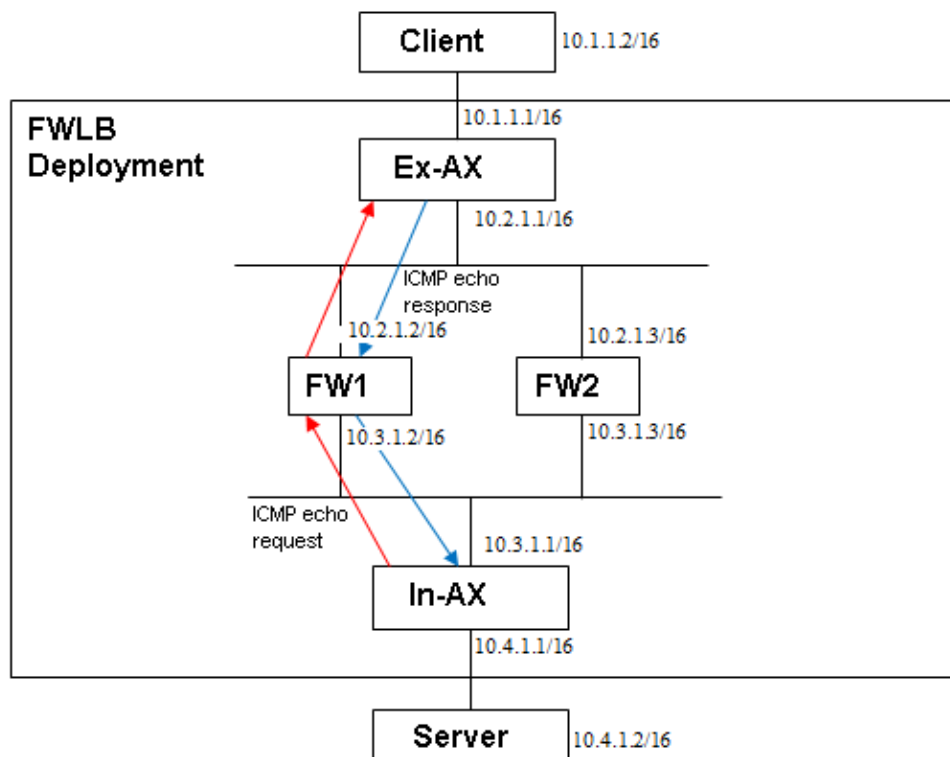
Health Monitoring for FWLB Paths

Optionally, you can configure each ACOS device to regularly test the paths through the firewalls to the other ACOS device. To test the paths, configure a Layer 3 health monitor (ICMP ping) and enable the “transparent” option.

The transparent option includes a special payload in the health-check packets that is recognized by the other ACOS device. For example, in [Figure 62](#), a health-check packet from “Ex-AX” travels through FW1 to “In-AX”. “In-AX” recognizes the payload and replies to the health check.

The red arrow in [Figure 62](#) below shows the path of this ICMP packet.

Figure 62 : Health monitoring for Firewall



In response, the external-facing ACOS device (“Ex-AX”) checks whether the ICMP echo request packet has a special payload. If the special payload is present, then “Ex-AX” sends an ICMP echo *response* packet to the source MAC address of “FW1”, which was contained in the original echo *request* packet. This ICMP echo response packet is represented by the blue arrow.

The health check (ICMP ping) occurs at Layer 3. The health checks at Layer 4 do not apply to FWLB ALG and must be disabled.

DNS Optimization

You can enable DNS caching globally or on individual VIPs.

The following topics are covered:

Overview	595
Configuring DNS Caching	607
DNS Recursive Resolver	659
DNS Over HTTP/HTTPS (DoH)	669
DNS Filtering Using Web Categories	673

Overview

ACOS can locally cache DNS replies and serve cached replies in response to DNS requests from clients. DNS caching offloads DNS servers, by caching replies to DNS queries and serving the cached replies in response to subsequent queries. This section describes the DNS optimization features supported by the ACOS device.

These features are supported only in software releases that support SLB. For information about DNS security features, see the *Application Access Management and DDoS Mitigation Guide*.

Global DNS Optimization

When DNS caching is enabled, the ACOS device forwards the initial request for a specific name, such as a hostname or fully qualified domain name, to the DNS server. Upon receiving a response, ACOS stores this information and subsequently provides the cached reply when additional requests for the same name are received.

Key Considerations

- ACOS continues to use the cached DNS reply until the reply times out. After the reply times out, ACOS sends the next request for the hostname to the DNS server, and caches the reply, and so on.
- DNS caching is disabled by default. When you enable it, replies are cached for a default duration of 300 seconds. Once a DNS reply is cached, the aging process starts immediately and keeps progressing, even when the cached response is accessed later. Accessing the cache does not refresh or reset the age.
- The maximum size for DNS cache entries can be 1-4096 bytes. The default is 256 bytes.
- Global DNS caching applies to DNS requests sent to a VIP with:
 - virtual-port type UDP (on port 53)
 - virtual-port-type DNS-UDP/DNS-TCP (on any port number).
- ACOS supports configuring a global maximum UDP DNS response size for dns-udp

type virtual ports that do not use a DNS template. This limit also applies to responses served from the DNS cache.

- This global value is overridden when the same option (`max-udp-size`) is configured at the DNS template level.
- The configured `dns-max-udp-size` value must be smaller than any `edns-udp-size` values defined without template-level `max-udp-size` in the same partition.
- If the original DNS response exceeds the allowed maximum size, ACOS returns a minimal response with the TC (truncated) flag set and instructs clients to retry the query over TCP.

NOTE: Global DNS cache also applies to the DNS requests that ACOS transfers by L2 switching or L3 routing. Global DNS caching for these requests is supported only for DNS requests sent over UDP (port 53). DNS caching is not supported for DNS requests sent to:

- other port numbers if the request is UDP
- any port number if the request is TCP

GUI Configuration

1. Navigate to **ADC>>SLB >> Global** tab.
2. Select the checkbox in the DNS Cache Enable field.
3. After selecting this checkbox, you can configure the Response Type and TTL Threshold.
4. Optionally, you can also configure the following global cache parameters:
 - DNS Cache Age
 - DNS Cache Entry Size
5. Click **Update** to save your settings.

CLI Configuration

- To enable DNS caching, use the following commands:

```
ACOS(config)# slb common
ACOS(config-common)# dns-cache-enable
```

- The default cache timeout is 300 seconds. To change the cache timeout to 500

seconds, use this command:

```
ACOS(config-common)# dns-cache-age 500
```

- To configure the global maximum UDP DNS response size, use this command:

```
ACOS(config-common)# dns-max-udp-size 1024
```

- The default maximum cache entry size is 256 bytes. This command changes the maximum size to 128 bytes.

```
ACOS(config-common)# dns-cache-entry-size 128
```

- To display global DNS caching statistical information, use the following command:

```
ACOS# show dns cache statistics
```

DNS Optimization per VIP

DNS caching per-VIP DNS provides finer control over DNS caching. You can configure caching policies to tightly control caching behavior.

- DNS caching on per-VIP basis
- DNS caching on per-record bases through DNS query-type filter
- Rate-based DNS caching
- DNS record weighting for selective cache entry aging
- Throttling based on domain name
- Logging of DNS cache hits

Parameters for DNS caching per VIP are configured in the following places:

- Class list
- DNS template
- Per-VIP DNS caching applies to DNS requests sent to a VIP with virtual-port type `dns-udp`, on any port number. DNS caching is not supported for DNS requests sent to virtual-port type `dns`, or requests sent over TCP.

For details on configuring DNS caching, see the `slb template dns` command in the *Command Line Interface Reference*.

Class-List Parameters

To control DNS caching based on domain name, you can use a class list. A class list used for DNS caching has entries in the following format:

```
match-option domain-string {glid | lid} num
```

Each entry consists of the following:

- *match-option* – Specifies match criteria for the *domain-string*. The *match-option* can be one of these:
 - *dns contains* – The entry matches if the DNS request is for a domain name that contains the *domain-string* anywhere within the requested domain name.
 - *dns starts-with* – The entry matches if the DNS request is for a domain name that begins with the *domain-string*.
 - *dns ends-with* – The entry matches if the DNS request is for a domain name that ends with the *domain-string*.
- *domain-string* – Specifies all or part of the domain name on which to match.
 - For wildcard matching on more than one character, you can use the *dns contains*, *dns starts-with*, and *dns ends-with* options. For example, “*dns ends-with example.com*” matches on both “*abc.example.com*” and “*www.example.com*”.
- *{glid | lid} num* – Specifies global limit ID (GLID) or local limit ID (LID). LIDs contain policies and are configured in individual DNS templates. GLIDs can be used by multiple DNS templates. GLIDs and LIDs contain DNS caching policies. ACOS applies DNS caching policy in the specified GLID or LID to the *domain-string*.
 - Each class list can contain a maximum of 8000 entries or 5000 *domain-string* characters.

Example Class List for DNS Caching

```
dns contains a10networks.com lid 1
dns starts-with www.example1.com lid 2
dns ends-with www.example2.com lid 3
dns contains example3 lid 4
```

Each class-list entry selects a DNS caching policy, contained in the LIDs, based on the matching hostname. For example, queries for hostnames that contain

“a10networks.com” are processed using the policy in LID 1. The LIDs are configured in a DNS template that is applied to the DNS virtual port. (See [DNS Template Parameters](#).)

DNS Template Parameters

You can configure the following DNS caching parameters in the DNS templates:

DNS caching state (DNS template enabled or disabled) – DNS caching is enabled by default. You can easily disable it by setting this parameter, without changing the configuration of other caching parameters or changing the configuration settings on the DNS virtual ports that use the template.

- Default caching policy – Specifies the cache action (cache or nocache) for requests that *do not* match any *domain-string* in the class list. The default is nocache. By default, replies for domain names that do not match the class list are not cached.
- Cache hit count - Enables the hit count to DNS cache entries. This can help reveal the number of hits made to the DNS cache. By default, the DNS cache hit count is disabled.
- cache-record-serving-policy - Specifies the cache record response policy for DNS template, which allows each service to have its own policy. The default is global. You can set round-robin and no-change options.
- TTL adjustment - When ACOS responds to a DNS query with the cached DNS response, the TTL value displayed is the same as the TTL value of the cached DNS response. To update the TTL value to represent the current TTL value, use the following command:

```
ACOS(config-dns)#cache-ttl-adjustment-enable
```

When the option is enabled, the TTL value is obtained using 'baseline value - lifetime in cache'. Baseline value is the smaller value from the following two values:

- Server response TTL value
- TTL value that is configured under the DNS template.

This is the TTL value that is configured in the classlist as LID, and this classlist is bound to the DNS template.

- Maximum cache size per vThunder device – Specifies the maximum number of entries that can be cached per VIP. The maximum configurable limit depends on

the RAM size allocated to vThunder. For CLI command `max-cache-size <num>`, replace `<num>` with a value as mentioned below:

- If RAM is less than or equal to 4G, the maximum number of entries that can be specified is 65535.
- If RAM is 4G to 16G, the maximum number of entries that can be specified is 512000.
- If RAM is 16G to 128G, the maximum number of entries that can be specified is 2097152.
- Maximum cache entry size – Specifies the maximum number of bytes a DNS cache entry can have. You can specify 1-4096 bytes. The default is 256 bytes.
- Maximum query length – Specifies the maximum number of bytes a DNS query can received by the DNS virtual port can contain. You can specify 1-4096 bytes. By default, this parameter is unset.
- Maximum UDP Payload Size – Specifies the maximum UDP payload size allowed for DNS responses sent to the client.
 - When the response size exceeds the configured value, ACOS sets the TC (truncated) flag in a minimal response and instructs the client to retry over TCP.
 - Configurable range: 512–4096 bytes
 - Applicable to all DNS responses handled through the template for dns-udp type virtual-port.
 - Must be less than or equal to `edns-udp-size` in recursive-dns-resolution configuration level of the same DNS template.
 - DNS queries and responses handled by GSLB are not supported for the maximum UDP payload size options.
- Recursive DNS Lookup - Enables ACOS to resolve DNS queries on behalf of the client when required. The following parameters can be configured:
 - An IPv4 or IPv6 NAT pool
 - A hostname AC type class-list
 - Retry behavior when a server responds with a `REFUSED` exit code for queries with EDNS(0) client subnet inserted. This option applies only when `insert-`

`client-subnet` is configured.

If a server responds with a **REFUSED** exit code for a query containing an EDNS(0) client subnet, ACOS retries the query without the inserted subnet.

NOTE: When recursive DNS lookup is enabled on a DNS template and bound to a virtual port, the virtual port is marked operationally up, even if all servers in the associated service group are down due to health check failures.

The Recursive DNS Lookup will be used in two scenarios:

- If there are no servers available to respond to client's DNS query (servers down due to health check)
- If the DNS server in the service-group bound to the virtual port responds to client's query without a matching record.

NOTE:

- Supports `dns-udp` and `dns-tcp` type virtual server ports only.
 - Supports caching the responses only when DNS caching is configured globally using `dns-cache-enable` command under `slb common` configuration mode.
 - Does not support the `query-id-switch` command when enabled on any DNS template in the same L3v partition.
-
- ACOS returns a minimal DNS response for recursive DNS lookup, by default. In this mode, only the Answer section is included, and the Authority and Additional sections are removed before the response is sent to the client. This behavior applies only to DNS responses and does not affect CLI output.
- Minimal DNS response is bypassed in the following scenarios:
- Negative response (NXDOMAIN and NODATA)
 - Delegation response

NOTE:

- The sections will not be removed for EDNS(0) as it follows bind server behavior.
 - The complete DNS response is stored in cache. When cache sharing is enabled, clients that do not have minimal DNS response behavior in effect may still receive full responses from the shared cache.
-
- When the **full-response** option is configured, ACOS serves all records (Authority and Additional sections) when applicable.
 - Lookup order can be specified for each DNS query type in recursive DNS lookup when both IPv4 and IPv6 nat-pools are configured. ACOS performs lookup via the preferred IP protocol. In case of failure, the lookup is performed with the secondary IP protocol.
 - Support to configure the maximum number of total trials that ACOS performs, cumulative of all levels based on per client DNS query, before quitting the recursive process.
 - Support to configure the maximum number of retries at each level of the DNS resolution process.
 - Add Padding to Client - Adds the EDNS(0) padding to allow client to pad DNS packets by a variable number of bytes, if the client-SSL template is configured. You can specify the block length and random block length values.
 - Remove Padding to Server - Removes the EDNS(0) padding to the server if the server-SSL is not configured.
 - Remove EDNS Subnet to Server - Remove EDNS (0) client subnet from the client queries.
 - Insert EDNS(0) Client Subnet - Insert Extension mechanism for DNS EDNS(0) Client Subnet (ECS) in the query (only if it is absent). This extension contains sufficient network information and allows a recursive DNS resolver to specify the network subnet for the host on whose behalf the DNS query is made. It also helps to speed up the delivery of data from content delivery networks by allowing better use of DNS-based load balancing. You can also specify the source prefix lengths to conceal a part of the IPv4 and IPv6 addresses.
- Although the implementation is based on [RFC 8781](#), the following features are not

implemented:

- Cache and scope related features
- Format checks to return FORMERR response, and to reject a client
- Actions related to REFUSED response
- Actions as an Authoritative name server and intermediate name server

NOTE: It is supported only on `dns-udp` and `dns-tcp` type virtual server ports only. Additionally, it is supported for DNS over HTTPs (DoH) as well.

- UDP Retransmit - Specify the policy to retransmit the UDP packet if the server does not respond after the specified time. You can specify the maximum trails and retry interval values.
- Logging – You can enable logging of DNS caching events. (See [DNS Cache Logging](#).) Logging is disabled by default. When you enable it, you can specify the number of minutes between generation of log messages, 1-10000 minutes.
- Class-list LID – Specifies the DNS policy (DNS caching settings) for domain-strings in the class list.
- Enable DNS64 - DNS64 supports the communication between IPv6 environment and IPv4 server. This feature provides the address transfer and also converts the request type if the server does not have the IPv6 interface to serve the client.

NOTE: This feature is not supported with DNS recursive resolution.

DNS Caching LID / GLID Policy Parameters

You can configure the following DNS caching policy settings either in a GLID or in a LID within the DNS template. Settings configured in a GLID can be used by multiple DNS templates, while settings within the LID in a DNS template can be used only by that template.

Options available when Global DNS Cache is disabled

The following options are available for configuring DNS cache using LID/GLID when Global DNS Cache is disabled:

- **DNS Caching** – Enabled or disabled
- **Connection rate limit** – Maximum DNS connection rate allowed before the over-limit action is applied. (See below.) You can specify 1-4294967295 DNS connections per 1-65535 100-millisecond (ms) intervals. There is no default value.
- **Over-limit action** – Action to take when the DNS connection or request limit or rate is exceeded. The default action is to drop the traffic. Optionally, you can specify one of the following actions instead:
 - Enable caching
 - Disable caching
 - Forward the traffic anyway
 - Lock out further traffic from the sender for the specified number of minutes, 1-1023. You also can specify a lockout time for any of the other over-limit actions listed above.

By default, logs are not generated by an over-limit action. You can enable logging of over-limit actions. Over-limit messages are sent immediately. They are not queued based on DNS cache logging settings.

- **Time-To-Live (TTL)** – Period an entry remains in the cache before it is removed. By default, the global DNS cache age is used. The default global DNS cache age is 300 seconds (5 minutes).
- **Weight** – Numeric value used when cache entries need to be removed to make room for new entries. You can assign a weight of 1-7. Lower-weighted objects are removed before higher weighted objects. The default weight is 1.
 - Cache more than 60% full, entries with weight 1 are eligible to be removed.
 - Cache more than 70% full, entries with weight 1 or 2 are eligible to be removed.
 - Cache more than 80% full, entries with weights 1-4 are eligible to be removed.
 - Cache more than 90% full, entries with weights 1-6 are eligible to be removed.

Options available when Global DNS Cache is enabled

The following parameters are available for configuring DNS cache using LID/GLID when Global DNS Cache is enabled:

- **Connection rate limit** - Configures the maximum number of connections allowed per second. The number of connections can range from 1-2147483647.
 - **per** - Configures the connection rate limit per second. The interval can range from 1-65535 seconds.
- **DNS Caching** - Configures DNS cache options. The following options are available:
 - **cache-disable** - Disables DNS cache.
If the DNS cache is enabled at the SLB common (global) configuration level, this LID configuration takes precedence.
 - **cache-enable** - Enables DNS cache.
 - **honor-server-response-ttl** - Enables TTL in the server response to be used as DNS cache TTL.
 - **ttl** - Configures TTL for cache entry.
 - **weight** - Configures weight for cache entry.
- **Over-limit action** - Configures the action when the connection rate limit is exceeded. The following options are available:
 - **ignore** - Ignores the connection rate limit and proceeds.
 - **log** - Enables system logs to generate log message for the ignored DNS query at the specified interval. The log interval can range from 1-255 minutes.
When logging is enabled, a separate message is generated for each over-limit occurrence every 1 minute, by default.
 - **drop** - Drops the DNS query when the connection rate limit exceeds.
 - **lockout** - Locks out the client for a specified number of minutes. During the lockout period, all connection requests from the client are dropped. The lockout period can be 1-1023 minutes. There is no default.
 - **log** - Enables system logs to generate log message for the dropped DNS query at the specified interval. The log interval can range from 1-255 minutes.
When logging is enabled, a separate message is generated for each over-limit occurrence every 1 minute, by default.

For more information on the Global DNS Cache configuration, see [Configure Global DNS Cache](#).

Additional Options (available only in GLIDs)

- **Connection limit** – Maximum active DNS connections allowed before the over-limit action is applied. You can specify 1-1048575 DNS connections. There is no default.
- **Request limit** – Maximum number of DNS requests before the over-limit action is applied. You can specify 1-1048575. There is no default.
- **Request rate limit** – Maximum rate of DNS requests before the over-limit action is applied. You can specify 1-4294967295 DNS requests every 1-65535 100-millisecond (ms) interval(s). There is no default.
- **Use NAT pool** – Binds a NAT pool to the GLID. The pool is used to provide reverse NAT for class-list members that are mapped to this GLID.

DNS Flag Day Compliance

The DNS Flag Day Compliance re-mediate any DNS servers and DNS resolvers that are not in compliance with EDNS. A variety of workarounds that were defined to accommodate DNS solutions that were not compliant have resulted in inefficiency and performance degradation. The following issues can be resolved through compliance check:

- Non-compliant solutions will either disappear from the public Internet or be compelled to be made compliant.
- DNS message fragmentation: If a DNS message is larger than the MTU of a path, the Layer 3 protocol (IPv4/IPv6) breaks apart the message into smaller packets.
- DNS messages sent through UDP are more prone to transmission errors when IP fragmentation occurs, and even in the best-case scenario there remains potential security hazards.
- DNS Server and resolver operators/vendors must set a default buffer size that will minimize, if not eliminate, fragmentation; and to ensure that the DNS solutions provided have adequate support for DNS over TCP, to address the shortcomings involved in fragmentation of DNS messages.
- To pass Flag Day 2020 for DNS “Authoritative” operators, the following settings are

required:

- Comply with RFC 7766 (mandating support for DNS over TCP)
- Adjust the buffer size to a tentative value of 1232
- Replies sent must not be larger than the buffer size
- The requirements for passing Flag Day for DNS “Resolver” operators are:
 - Comply with RFC 7766
 - Adjust the EDNS buffer size
 - Fall back to TCP in case of DNS truncation over UDP

Configuring DNS Caching

This section includes the DNS caching related configuration instructions and examples.

The following topics are covered:

Overview	608
Configure a Class List	608
Configure the GLIDs	609
Configure a DNS Template	610
Enable DNS Caching on a VIP	611
Display and Clear DNS Caching Information	612
Display DNS Cache Hits and DNS Query Rate	613
Synchronize Global DNS Cache	614
Configure Global DNS Cache	615
Enable Persistent Storage for Global DNS Cache	618
Configure DNS Cookie Cache Policy	620
Configure Negative Caching of DNS Response	621
DNS Cache Logging	625
Enable Hit Count of DNS Cache Entries	625
DNS Logging	626

DNS Queries per Second Logging	643
Authenticate DNS Requests	646
Configuration Examples – DNS Optimization	647
Load Balancing with the “DNSSEC OK” (DO) Bit	657

Overview

To configure per-VIP DNS caching:

- Configure a class list that maps domain-strings to GLIDs or LIDs.
- If using GLIDs to specify the DNS caching policies, configure the GLIDs.
- Configure a DNS template. Specify the name of the class list. Bind the class list to the GLIDs, or configure the LIDs under the class list.
- Configure a real server for the DNS server. Add UDP port 53 to the real server.
- Configure a UDP service group and add the real server to it.
- Configure a VIP using a virtual port with service type dns-udp. Bind the DNS template and the service group to the virtual port.
- Service groups do not support stateless load-balancing methods with DNS features listed in this section.

You can configure a class list in either of the following ways:

- Use a text editor on a PC or other device to create the list, then import it onto the ACOS device.
- Use CLI commands to create the list entries.

For class-list syntax information, see [Class-List Parameters](#).

Configure a Class List

Importing a Class List onto the ACOS device

To import a class list onto the ACOS device, use the `import class-list` command at the Privileged EXEC or global configuration level. The command specifies the name

the class list will have on the ACOS device along with the file transfer protocol, username (when required), and directory path.

- You can enter the entire URL on the command line or press Enter to display a prompt for each part of the URL. If you enter the entire URL and a password is required, you will still be prompted for the password. To enter the entire URL:

- **tftp:** //host/file
- **ftp:** //[user@]host[:port]/file
- **scp:** //[user@]host/file
- **sftp:** //[user@]host/file

You also can export class lists to a remote server, using the **export class-list** command.

Deleting a class list from a file system is the same as saving the configuration changes to the file system. The **write mem** command is required.

Configuring a Class List in the ACOS Device

To configure a class list in the CLI, use the **class-list** command. You can either save the class list as a separate file or save it to the startup-config. If the class list contains 100 or more entries, it is recommended to save the list to a file. A class list can be exported only if you use the **file** option.

The **class-list** command creates the class list if it is not already configured, and changes the CLI to the configuration level for the list, where the **dns** command is available. Use the command to configure match rules to map DNS traffic to LIDs or GLIDs. (See [Class-List Parameters](#).)

Configure the GLIDs

To configure DNS caching policies in GLIDs, use the following methods to configure each GLID. To use LIDs configured within the DNS template, go to [Configure a DNS Template](#).

Use the **glid** command at the global configuration level. This command creates the GLID and changes the CLI to the configuration level for it, where the following commands are available.

- The `conn-limit` command specifies the maximum number of concurrent connections on the DNS virtual port before the over-limit action is applied.
- The `conn-rate-limit` command specifies the maximum DNS connection rate before over-limit action is applied.
- The `over-limit-action` command specifies the action when the DNS connection, request limit, or rate is exceeded.
- The `request-limit` command specifies the maximum number of DNS requests before over-limit action is applied.
- The `request-rate-limit` command specifies the maximum DNS request rate before over-limit action is applied.
- The `dns` command disables or enables DNS caching.
- The `dns ttl` command specifies the number of seconds to keep an entry in the cache before removing it.
- The `dns weight` command specifies the numeric value used for removing cache entries to make room for new entries.

Configure a DNS Template

Configure a DNS Template (CLI Procedure)

To configure a DNS template for DNS caching, enter the `slb template dns` command at the global configuration level of the CLI. The command creates the template and changes the CLI to the configuration level for it, where the following commands are available.

For configurable ranges and default values, see [DNS Template Parameters](#).

- The `default-policy` command specifies the default action for DNS queries for domain names that do not match a class list entry.
- The `dns-log-enable period` command enables logging and specifies how often to generate logs.
- The `max-cache-size` command specifies the maximum number of replies to cache per DNS virtual port.
- The `class-list name` command specifies the name of the class list to use.

- The `class-list lid` command creates a LID and changes the CLI to configure it. LID numbers are from 1 to 1023.

To disable the DNS template without removing the template from the configuration, use the `disable-dns-template` command:

LID Commands for DNS Caching (CLI Procedure)

The following commands are available at the configuration level for DNS caching LIDs.

- The `conn-rate-limit` command specifies the maximum DNS connection rate before over-limit action is applied.
- The `over-limit-action` command specifies the action when DNS connection, request limit, or rate is exceeded.
- The `dns cache-disable` command disables DNS caching. The `dns cache-enable` command enables caching.
- The `dns ttl` command specifies the number of seconds to keep an entry in the cache before removing it.
- The `dns weight` command specifies the value used when cache entries are removed for space for new entries.

Enable DNS Caching on a VIP

Enable DNS Caching on a VIP (CLI Procedure)

To enable DNS caching on a Virtual IP (VIP), configure the VIP to use a `dns-udp` and `dns-tcp` service type. DNS caching is supported on virtual ports configured with the `dns-udp` and `dns-tcp` service type.

1. Configure the virtual server port

At the virtual server configuration level, use the `port` command to add a `dns-udp` port. This step changes the CLI context to the virtual port level.

2. Bind the DNS template

Within the virtual port configuration context, use the `template dns` command to attach the appropriate DNS template. This template enables DNS caching and related DNS functionality for the VIP.

3. Associate a service group

Ensure that a UDP or TCP based service group is bound to the virtual port.

The configuration for real servers and service groups is identical to other SLB virtual port configurations and is therefore not repeated in this section.

Display and Clear DNS Caching Information

This section describes how to view and clear DNS cache configuration information and statistics using the CLI and the GUI.

Using the CLI

To view complete information on the show and clear DNS cache commands, including syntax, options, filters, and usage examples, see the `show dns cache` and `clear dns cache` commands in the *Command Line Interface Reference*.

Using the GUI

Go to **ADC > Statistics > L7 > DNS > DNS Cache Entry**. The following options can be selected to view the statistics on this page:

- **Global** - The filter is applied to DNS cache global entries.
- **Virtual Server** - This field is enabled only when 'Global' field is disabled. Select the virtual server from the list.
- **DNS Virtual Port** - This field is enabled only after 'Virtual Server' field is enabled. Then, select the virtual port from the list.
- **Content Mode** - Select one of the following from the list:
 - **Brief** - Displays the DNS response resource records of the cache entry in brief. It displays only the answer section.
 - **Detail** - Displays the DNS response resource records of the cache entry in detail. It displays the answer section, authority section and additional section.

The following options are enabled when **Content Mode** is specified:

- **Max Record Number** - Select 'All' or specify a value between 1-1024.
- **Max Record Data Size** - Select 'Full' or specify a value between 1-8192.
- **DNS Class** - Select one of the options from the list or specify a value between 1-65535.
- **DNS Type** - Select one of the options from the list or specify a value between 1-65535.
- **Domain Name** - Select from the list (**DNS Domain** or **FQDN Domain**) and specify the **Domain Name**.

Display DNS Cache Hits and DNS Query Rate

When DNS cache is enabled, the rate of current DNS cache hits and DNS Queries-Per-Second (QPS) can be displayed on-demand using the `show dns cache rate` command. It displays the following kind of response:

```
Query rate(QPS) : 50 per second
Cache hit       : 41 per second
Cache hit ratio : 82% (cache hit/Query)
```

QPS traces DNS query volumes and DNS cache hit ratio gives an insight into cache performance. You can achieve a real time performance insight with SNMP OID by combining this data. For more information on SNMP OIDs, see MIBs Reference Guide.

Limitation

- While retrieving the current DNS QPS and DNS cache hit rate, the CLI momentarily pauses for one second. This brief pause halts the processing of CLI commands to gather real-time and accurate data on DNS QPS and DNS cache hit rate.
- For multi-PU platforms, the cache hit ratio counter may not function as expected.
- QPS, cache-hit rate and cache hit ratio values displayed in aXAPI/SNMP and CLI will be different or may not match with each other.
- All the counters in the SNMP table (dnsCacheOperTable) are currently not supported, except the cache hit ratio. Use appropriate filter in the aXAPI to retrieve the specific details.

Synchronize Global DNS Cache

Enabling DNS cache optimizes CPU utilization on the active ACOS device because most DNS requests get served directly through the cache. However, a failover event can cause CPU usage to spike on the standby ACOS device because the device has to resolve all destination hosts in bulk. This CPU spike becomes amplified if recursive DNS service is also enabled.

To alleviate this issue, you can configure global DNS cache entry synchronization. Upon failover, the standby device stays ready to resolve DNS queries from cache.

NOTE: The synchronized messages that are lost cannot be recovered.

The following configurations can be done after the DNS cache synchronization is enabled:

- You can configure DNS cache entries with smaller size to be synchronized to the standby device. The default value is 256 bytes, and the configurable range is 1-4096.
- You can configure DNS cache with higher TTL value to be synchronized to the standby device. The default value is 0 seconds, and the configurable range is 0-10000000.

CLI Configuration

To configure global DNS cache synchronization, use the following commands:

```
ACOS(config)# slb common
ACOS(config-common)# dns-cache-sync
ACOS(config-common)# dns-cache-sync-entry-size 500
ACOS(config-common)# dns-cache-sync-ttl-threshold 300
```

NOTE: `dns-cache-sync` feature only works for default vrid (vrid-0) for global DNS cache.

Configure Global DNS Cache

Enabling global DNS cache allows ACOS to manage DNS cache efficiently and optimize the DNS resolution. The global cache configuration uses the class-list configurations and LID/GLID mapping to apply specific policies or limits to the incoming traffic. It allows DNS cache bypass based on domain names and/or source IPv4/IPv6 address, enables connection rate limit based on limit identifiers (LID/GLID), and provides TTL (Time-to-Live) entry management capabilities.

NOTE: When the ACOS device is running recursive resolution, DNS responses from internet servers during resolution and/or populated Name Server and A/AAAA cache entries are not supported when global DNS cache is enabled.

To configure global DNS cache using LID:

1. Create a class-list of type DNS (domain-name based) and/or IPv4/IPv6 address (source IP based) using LID:

```
ACOS(config)# class-list cl_dns_cache
ACOS(config-class list)# dns contains www1.test.com lid 1
ACOS(config-class list)# dns contains www2.test.com lid 2
```

2. Enable DNS cache globally:

```
ACOS(config)# slb common
ACOS(config-common)# dns-negative-cache-enable
ACOS(config-common)# dns-cache-enable
ACOS(config-common)# dns-cache-ttl-adjustment-enable
```

3. Define policies for each LID/GLID within the global DNS cache:

```
ACOS(config-common)# global-dns-cache
ACOS(config-common-global-dns-cache)# class-list cl_dns_cache
ACOS(config-common-global-dns-cache-class...)# lid 1
ACOS(config-common-global-dns-cache-class...)# dns cache-disable
ACOS(config-common-global-dns-cache-class...)# user-tag dns_cache_bypass
ACOS(config-common-global-dns-cache-class...)# exit
ACOS(config-common-global-dns-cache-class...)# lid 2
ACOS(config-common-global-dns-cache-class...)# dns cache-enable
ACOS(config-common-global-dns-cache-class...)# dns ttl 1000
ACOS(config-common-global-dns-cache-class...)# user-tag entry_manual_ttl
ACOS(config-common-global-dns-cache-class...)# exit
ACOS(config-common-global-dns-cache-class...)# exit
ACOS(config-common-global-dns-cache)# exit
ACOS(config-common)# exit
```

4. Configure the real servers:

```
ACOS(config)# slb server rs1 192.168.1.1
ACOS(config-real server)# port 53 tcp
ACOS(config-real server)# port 53 udp
ACOS(config-real server)# exit
ACOS(config)# slb server rs2 192.168.1.2
ACOS(config-real server)# port 53 tcp
ACOS(config-real server)# port 53 udp
ACOS(config-real server)# exit
```

5. Configure service groups for the real servers:

```
ACOS(config)# slb service-group sgt tcp
ACOS(config-slbg svc group)# member rs1 53
ACOS(config-slbg svc group-member:53)# exit
ACOS(config-slbg svc group)# member rs2 53
ACOS(config-slbg svc group-member:53)# exit
ACOS(config-slbg svc group)# exit
ACOS(config)# slb service-group sgu udp
ACOS(config-slbg svc group)# member rs1 53
ACOS(config-slbg svc group-member:53)# exit
ACOS(config-slbg svc group)# member rs2 53
ACOS(config-slbg svc group-member:53)# exit
ACOS(config-slbg svc group)# exit
```

6. Configure a virtual server and port on the ACOS device and associate them with the proper service group:

```
ACOS(config)# slb virtual-server vs 10.211.1.100
ACOS(config-slbg vserver)# port 53 dns-tcp
ACOS(config-slbg vserver-vport)# service-group sgt
ACOS(config-slbg vserver-vport)# exit
ACOS(config-slbg vserver)# port 53 dns-udp
ACOS(config-slbg vserver-vport)# service-group sgu
ACOS(config-slbg vserver-vport)# exit
```

To configure global DNS cache using GLID:

1. Define policies for GLID:

```
ACOS(config)# glid 1
ACOS(config-glid:1)# conn-rate-limit 1 per 50
ACOS(config-glid:1)# over-limit-action drop
ACOS(config-glid:1)# exit
```

2. Create a class-list of type DNS (domain-name based) and/or IPv4/IPv6 address (source IP based):

```
ACOS(config)# class-list cl_dns_cache
ACOS(config-class list)# dns contains www.test.com glid 1
```

3. Enable DNS cache globally:

```
ACOS(config)# slb common
ACOS(config-common)# dns-cache-enable
ACOS(config-common)# dns-cache-age 600
ACOS(config-common)# dns-cache-entry-size 4096
```

4. Configure class-list within the global DNS cache:

```
ACOS(config-common)# global-dns-cache
ACOS(config-common-global-dns-cache)# class-list cl_dns_cache
```

5. Configure the real servers:

```
ACOS(config)# slb server rs1 192.168.1.1
ACOS(config-real server)# port 53 tcp
ACOS(config-real server)# port 53 udp
ACOS(config-real server)# exit
ACOS(config)# slb server rs2 192.168.1.2
ACOS(config-real server)# port 53 tcp
ACOS(config-real server)# port 53 udp
ACOS(config-real server)# exit
```

6. Configure service groups for the real servers:

```
ACOS(config)# slb service-group sgt tcp
ACOS(config-slbg svc group)# member rs1 53
ACOS(config-slbg svc group-member:53)# exit
ACOS(config-slbg svc group)# member rs2 53
ACOS(config-slbg svc group-member:53)# exit
ACOS(config-slbg svc group)# exit
ACOS(config)# slb service-group sgu udp
ACOS(config-slbg svc group)# member rs1 53
ACOS(config-slbg svc group-member:53)# exit
ACOS(config-slbg svc group)# member rs2 53
ACOS(config-slbg svc group-member:53)# exit
ACOS(config-slbg svc group)# exit
```

7. Configure a virtual server and a port on the ACOS device and associate them with the proper service group:

```
ACOS(config)# slb virtual-server vs 10.211.1.100
ACOS(config-slbg vserver)# port 53 dns-tcp
ACOS(config-slbg vserver-vport)# service-group sgt
ACOS(config-slbg vserver-vport)# exit
ACOS(config-slbg vserver)# port 53 dns-udp
ACOS(config-slbg vserver-vport)# service-group sgu
ACOS(config-slbg vserver-vport)# exit
```

Enable Persistent Storage for Global DNS Cache

The ACOS DNS optimization feature includes persistent storage capability to store cached DNS entries. When the ACOS device needs to be serviced or restarted, the

cached entries are loaded from persistent storage. This improves the performance of the CPU and effectively manages the cached entries.

This feature can be enabled using `dns-persistent-cache-enable` command. The following configurations can be done after the persistent storage feature is enabled:

- You can limit the maximum persistent cache entries. The command can be configured only in a shared partition. The maximum value is the maximum supported DNS cache on the platform.
- You can set the TTL threshold value for persistent cache. This saves the DNS cache entries with higher TTL value. The value can be set from 0-10000000. The default value is 0.
- You can set the hit count threshold for persistent cache. This saves the DNS cache entries with higher hit count. The value can be set from 0-10000000. The default value is 0.

CLI Configuration

Before enabling persistent storage feature, make sure to enable DNS cache using `dns-cache-enable` command.

To configure persistent storage for global DNS cache, the following commands are introduced under SLB common mode:

```
ACOS(config)#slb common
ACOS(config-common)# dns-cache-enable
ACOS(config-common)# dns-persistent-cache-enable
ACOS(config-common)# max-persistent-cache 1024
ACOS(config-common)# dns-persistent-cache-ttl-threshold 200
ACOS(config-common)# dns-persistent-cache-hit-threshold 5
ACOS(config-common)# dns-cache-entry-size 4096
```

Show Command

- To view the persistent cache entries, use the following command:

```
ACOS#show dns persistent-cache entry
S = DNSSEC, T = Type, C = Class
Qlen = Query length, Rlen = Response length
Domain                               S    T    C    Qlen  Rlen  TTL
-----+-----+-----+-----+-----+-----
```

test.dnsrr.com	0	2	1	32	90	300
www.test.dnsrr.com	0	1	1	36	98	300
ns3.test.dnsrr.com	0	1	1	36	26	300
Total: 3						

- To view the persistent cache statistics, use the following command:

```
ACOS#show dns persistent-cache statistics
```

```
Total
```

```
-----
```

```
Total persistent cache 3
```

```
Total saved 3
```

```
Total deleted 0
```

```
Database Busy 0
```

```
Database Error 0
```

Known Limitation

- The DNS cache that belongs to the virtual port is not maintained.
- When many cache entries are added or deleted from the persistent storage, and the DNS queue is full, the DNS entries may not sync to the persistent storage.
- When the DNS cache is loaded, the persistent storage database does not refresh the cache age. If the cache is expired, the ACOS device deletes the expired cache from the persistent storage.

Configure DNS Cookie Cache Policy

During DNS caching, the presence of a cookie record in a DNS request can potentially lead to issues depending on the client's DNS cookie policy. The DNS cookie is used as a security check to ensure that the client IP, server IP, and content have not been altered during transmission (See [RFC 7873](#)).

The DNS Cookie Cache policy allows you to filter the cookie records. This policy can be configured to either serve DNS queries from the cache without including cookie records or to forward these queries to the backend server, ensuring that the cookie records are included in the response.

Using the CLI

For client queries with cookie records, ACOS presents the following options at the Global level and virtual IP level within the SLB template to enable clients to better manage DNS requests with cookie records:

Global DNS Cache

1. [Configure Global DNS Cache](#).
2. Set the cookie cache policy in `dns-cookie-cache-policy`. The policy can be set to serve queries either from the cache or be served by the DNS backend server.
 - **served-by-cache**: Indicates that DNS queries will be served using cached content. If a cache hit occurs, the response will not include a cookie record. If the cache is not hit, the response; including the cookie record, will be determined by the backend server settings.
 - **served-by-backend**: Indicates that DNS queries, regardless of cache status, will be forwarded to the backend server. Responses will include cookie records based on the backend server's settings.

Per Virtual IP DNS Cache

1. For virtual IP-level caching policy in the SLB DNS templates, configure [DNS Template Parameters](#).
2. Set the cookie cache policy in `dns-cookie-cache-policy`. Similar to the global setting, this option determines the caching policy for DNS queries, at the virtual IP level in the SLB template DNS.
 - **served-by-cache**: Indicates that DNS queries for this virtual IP will be served using cached content. Cached responses will not include cookie records.
 - **served-by-backend**: Indicates that DNS queries for this virtual IP will be forwarded to the backend server regardless of the cache status. Responses will include cookie records based on the backend server's settings.

Configure Negative Caching of DNS Response

ACOS supports negative caching or caching the negative responses for a domain. A negative response indicates that the requested domain information is unavailable or

that the server is unable to respond to the query. This feature enhances query responses while decreasing backend DNS traffic.

A negative response could be one of the following:

- **NXDOMAIN** - Indicates "Name Error" response, which means the domain name does not exist. For example,

```
Header:
  RDCODE=NXDOMAIN
Query:
  AN.EXAMPLE. A
Answer:
  AN.EXAMPLE. CNAME TRIPPLE.XX.
Authority:
  XX. SOA NS1.XX. HOSTMASTER.NS1.XX. ....
  XX. NS NS1.XX.
  XX. NS NS2.XX.
Additional:
  NS1.XX. A 127.0.0.2
  NS2.XX. A 127.0.0.3
```

- **NODATA** - Indicates "NOERROR" response, which means records for the requested domain name exist but do not match the record type. For example,

```
Header:
  RDCODE=NOERROR
Query:
  ANOTHER.EXAMPLE. A
Answer:
  <empty>
Authority:
  EXAMPLE. SOA NS1.XX. HOSTMASTER.NS1.XX. ....
  EXAMPLE. NS NS1.XX.
  EXAMPLE. NS NS2.XX.
Additional:
  NS1.XX. A 127.0.0.2
  NS2.XX. A 127.0.0.3
```

DNS caches negative responses based on the following types of negative entries:

- Valid- negative responses: These responses include SOA records in the DNS query response. For example,
 - CNAME and SOA records
 - NS and SOA records
 - Empty response and SOA records
- Non-valid negative responses: These responses do not include SOA records in the DNS query response.

The TTL for non-valid negative responses is 300 seconds.

NOTE:

- For both valid and non-valid negative responses, the `show dns cache entry` command displays 'N' as 1.
 - Non-valid negative cache will not be served to DNS queries.
-

CLI Configuration

Before enabling the negative caching feature, configure the `dns-cache-enable` global command. Also, ensure that the DNS and virtual servers are functional.

- To enable negative caching globally, only valid negative responses are cached by default. Use the following command under SLB common:

```
ACOS(config)# slb common  
ACOS(config-common)# dns-negative-cache-enable
```

- To enable both valid and non-valid negative caching globally, use the following command under SLB common:

```
ACOS(config)# slb common  
ACOS(config-common)# dns-negative-cache-enable cache-non-valid
```

- To configure the threshold for bypassing negative cache responses globally, use the following command under SLB common:

```
ACOS(config)# slb common  
ACOS(config-common)# dns-negative-cache-bypass-threshold 200
```

The bypass threshold value ranges from 1 to 65,535, with a default of 100.

- To configure valid and non-valid negative caching per virtual port (dns-tcp and dns-udp), use the following commands under SLB template DNS:

Bind the DNS template to the virtual port.

```
ACOS(config)# slb template dns dns_negative
ACOS(config-dns)# negative-dns-cache enable
ACOS(config-dns-negative-dns-cache)# bypass-query-threshold 200
ACOS(config-dns-negative-dns-cache)# max-negative-cache-ttl 1024
ACOS(config-dns-negative-dns-cache)# cache-non-valid
ACOS(config-dns-negative-dns-cache)# exit
ACOS(config-dns)# exit
ACOS(config)# slb virtual-server vip-cache 192.168.10.10
ACOS(config-slb vserver)# port 53 dns-udp
ACOS(config-slb vserver-vport)# template dns dns_negative
```

The negative response also includes TTL like standard response, which is taken from the SOA record. When the TTL reaches zero, the DNS server no longer uses the cached negative response.

However, the negative response may continue to loop between the two servers indefinitely, even if the TTL is short. To mitigate this, the ACOS device serves the negative response from the cache until the bypass query threshold (default 100) is reached. Similarly, the maximum TTL period is set to 2 hours by default.

Show Command

To view the global negative cache entries, use the following show command:

```
ACOS #show dns cache entry
S = DNSSEC, T = Type, C = Class, W = weight
Qlen = Query length, Rlen = Response length, N = Negative
Domain          N      S      T      C      Qlen  Rlen  TTL      Age      W      Hit
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----
Global Table
rirns.arin.net   1 0    28  1    32    38    172800  0        1  0
rirns.arin.net   0 0     1  1    32    26    172800  0        1  1
```

To view the per VIP negative cache entries, use the following command:

```
ACOS #show slb virtual-server vs 53 dns-udp application-statistics | begin
Negative
Negative Cache Served:      2
```

DNS Cache Logging

Logging for DNS caching is disabled by default. When you enable it, the following information is logged:

- Number of DNS cache hits between log intervals or before aging time
- ACOS management IP address, severity level (6, informational) and domain name requested

The hits counter is reset to 0 after the messages are sent. The counter is not cumulative.

Here are some examples:

```
Aug 22 04:05:14 192.168.20.164 a10logd: [DNS-Cache]<6> dns=example.com
hits=6 for the last 1 minute interval
Aug 22 04:05:16 192.168.20.164 a10logd: [DNS-Cache]<6> dns=example5.nl
hits=5 for the last 1 minute interval
```

Enable Hit Count of DNS Cache Entries

Recording the hit count of DNS cache entries is disabled by default. The following cases list the exceptions, where the hit count of DNS cache entries is enabled.

- Logging for DNS cache is enabled: In dns template, if the `dns-log-enable` command is configured, it keeps generating a log for each cache entry after the specified period. The log generation considers the cache hit count. However, after the log is generated, the hit count gets cleared. This function only supports the per-virtual port cache, and not the global cache.
- Negative DNS cache is enabled: If negative dns cache is enabled for negative cache entries, the cache hit count option is enabled. The bypass-query threshold of negative entry is based on the hit count to the cache entry. It determines whether a DNS query should be served from the cache or forwarded to the device. The negative DNS cache can be configured at the global level as well as for per-virtual port.
- Enabling hit count of DNS cache entries: The hit count of DNS cache entries can be enabled separately at the global level as well as for per-virtual port cache.

CLI Configuration

Before enabling the cache hit count feature, configure the `dns-cache-enable` global command. Also, ensure that the DNS and virtual servers are functional.

- To enable hit count to global DNS cache entries, use the following command:

```
ACOS(config)#slb common
ACOS(config-common)#dns-cache-hitcount-enable
```

The number of hits to each cache entry can be displayed in the `show dns cache entry` command.

- To enable hit count to DNS cache entries at the per virtual port level, use the `cache-hitcount-enable` command in slb template dns. The following sample shows the configuration in the `slb template dns` command and binds the DNS template to the virtual port.

```
ACOS(config)#slb template dns tempcache1
ACOS(config-dns)#cache-hitcount-enable
ACOS(config-dns)# exit
ACOS(config)#slb virtual-server dns-cachehit 192.168.20.10
ACOS(config-slb vserver)#port 53 dns-tcp
ACOS(config-slb vserver-vport)#service-group dns-tcp-53-1
ACOS(config-slb vserver-vport)#template dns tempcache1
```

DNS Logging

ACOS supports DNS logging through a template-based configuration that provides lightweight, flexible, and customizable output. DNS logging templates can be created, updated, or disabled to meet operational requirements.

All DNS logs are sent only to remote log servers. The remote logging destination must be configured using `acos-events`. For more information on `acos-events`, see *Event Logging Guide*.

Overview

ACOS provides two DNS logging methods to support different operational needs, one for simplified standard logging and another for detailed custom logging.

DNS log events are generated only when standard or custom logging options are explicitly configured in the DNS logging template.

A log entry is generated based on one of the following trigger reasons:

- Request – A DNS query is received.
- Response – A DNS reply is sent back to the client.
- Timeout – No response was received within the configured timeout threshold (supported only with custom logging).

DNS Logging Types

Standard Logging

Standard logging provides a simplified and predefined logging method which can be used when lightweight DNS visibility is sufficient.

It supports:

- Protocol-specific logging (TCP, UDP, or both)
- Selective section-level logging (header, question, answer, all)
- Record-type controls and length limits for DNS response RDATA fields
- RPZ filtering for DNS request events

Standard Logging Options

To configure a standard DNS logging template, use the following options with `s1b template dns-logging` command:

Option	Sub-Option/Value	Description
protocol	both	Logs DNS over both TCP and UDP.
	tcp	Logs DNS over TCP only.
	udp	Logs DNS over UDP only.
request-section	all	Logs DNS header and question sections.
	header	Logs DNS header section only.
	question	Logs DNS question section only.

Option	Sub-Option/Value	Description
response-include-rcode	—	Logs RCODE in the DNS response log.
response-section	all	Logs DNS response header, answer, authority, and additional sections.
	header	Logs DNS response header only.
	answer	Logs DNS response header and answer section.
response-tuple	client-side	Logs client-side response tuple information.
	server-side	Logs server-side response tuple information.
response-type	config	Configures the RDATA type for response logging. The following option is available: <code>type <record-type></code> - Specifies the response record type. After selecting a record type, you can configure its additional parameters such as <code>length-limit</code>. The <code>length-limit</code> can be set per record type in the range 0–256 bytes. For the <code>all</code> option, the maximum allowed <code>length-limit</code> is 1024 bytes. Supported <code>record-types</code> are <code>TXT</code>, <code>RRSIG</code>, <code>TSIG</code>, <code>DNSKEY</code>, <code>DS</code>, <code>CAA</code>, <code>NAPTR</code>, <code>OPT</code>.
standard-log	request	Logs standard DNS request information. The following option is available: <code>log-filter RPZ</code> - Sets log filter for RPZ request type only.

Option	Sub-Option/Value	Description
type	query	Logs DNS queries only.
	response	Logs DNS responses only.
	both	Logs both DNS queries and responses.

Standard Log Format

When standard logging options are configured, logs are generated based on the predefined DNS events. The format of DNS log messages depends on the remote logging infrastructure configured through `acos-events` (SYSLOG or CEF).

Custom Logging

Custom logging provides granular and highly configurable DNS log generation which can be used when detailed event visibility or precise control over log content and structure is required.

It supports:

- User-defined format strings
- Keyword-based field selection
- RPZ filtering for request events

Custom Logging Options

To configure a custom DNS logging template, use the following options with `slb template dns-logging` command:

Option	Sub-Option/Value	Description
custom-log request	enable	Enables custom logging for DNS requests.
	format <format_string>	Configures user-defined format string using logging keywords. For more information, see Custom Logging Keywords .
	log-filter RPZ	Logs only RPZ request events.

Option	Sub-Option/Value	Description
custom-log response	enable	Enables custom logging for DNS responses.
	format <format_string>	Configures user-defined format string using logging keywords. For more information, see Custom Logging Keywords .
custom-log timeout	enable	Enables custom logging for DNS timeout events. For more information, see Timeout Logging .
	format <format_string>	Configures user-defined format string using logging keywords. For more information, see Custom Logging Keywords .

Custom Log Format

When custom DNS logging is configured, logs are generated based on the custom format string and the selected keywords. The format of DNS log messages depends on the remote logging infrastructure configured through `acos-events` (SYSLOG or CEF).

Format String Rules

- A valid, non-empty format string is required before enabling custom logging.
- The format string cannot be removed or left blank.
- The format string must be less than 1024 bytes. If this limit is exceeded, ACOS generates a warning and logs the event.
- Maximum allowed CEF field specification is cs6/cn4.

SYSLOG Format

```
<timestamp> <log-level> [ACOS]: <log content>
```

CEF Format

```
<timestamp> <log-level> [ACOS]: vThunder
CEF:0|A10|ADC|<version>|<event-id>|Custom DNS <trigger reason>
log|<severity>|<log content>
```

- `<log content>` is fully controlled by the custom logging keywords and format string.
- `<trigger reason>` may be Request, Response, or Timeout.

Timeout Logging

A timeout log is generated when:

- A DNS server does not respond to a client query within the configured timeout window.
- ACOS terminates the connection after exceeding the retry or time threshold defined in the associated UDP or TCP template.
- The client terminates the session prematurely due to timeout or retry exhaustion.

Custom Logging Keywords

Custom DNS logging uses predefined keywords that represent individual DNS fields. These keywords can be inserted into the custom-log format string to control the content and structure of log messages.

Keyword Rules

- Each keyword corresponds to a specific DNS field from a request, response, or timeout event.
- Keywords must be enclosed in `$...$` symbols.
- A keyword may appear only once per trigger reason, duplicate usage is not allowed.
- Not all keywords apply to all event types (request, response, timeout).

Supported Keywords

To view the supported custom logging keywords, use the `show slb dns-custom-logging-keywords <trigger reason>` command.

For keywords specific to each trigger reason, see this command in the *Command Line Interface Reference*.

The following table lists all the supported keywords for custom logging:

Keyword	Description	Example Output in logs
<code>\$queryid\$</code>	Query ID used in the DNS logging.	2BD8
<code>\$opcode\$</code>	Opcode used in the DNS logging.	0-F
<code>\$headerflag\$</code>	Header flag used in the DNS logging.	RD RA
<code>\$recordcounters\$</code>	Combination of qdcount, ancourt, nscount, arcount used.	1 1 0 0
<code>\$qdcount\$</code>	Number of questions (QDCOUNT) used in the DNS logging.	1
<code>\$ancount\$</code>	Number of answers (ANCOUNT) used in the DNS logging.	1
<code>\$nscount\$</code>	Number of authority records (NSCOUNT) used in the DNS logging.	0
<code>\$arcount\$</code>	Number of additional records (ARCOUNT) used in the DNS logging.	0
<code>\$respsrc\$</code>	Response source information used.	gslb/cache/backend/rpz
<code>\$queryname\$</code>	DNS query name used.	dns.example.com
<code>\$querytype\$</code>	DNS query type used.	A
<code>\$queryclass\$</code>	DNS query class used.	IN
<code>\$queryinfo\$</code>	Combination of information with query name, query type, query class used.	dns.example.com A IN

Keyword	Description	Example Output in logs
<code>\$rcode\$</code>	DNS response (rcode) used.	0000-FFFF
<code>\$rcodestr\$</code>	DNS response code (string format).	0 NoError No Error [RFC 1035] 1 FormErr Format Error [RFC 1035] 2 ServFail Server Failure [RFC 1035] 3 NXDomain Non-Existent Domain [RFC 1035] 4 NotImp Not Implemented [RFC 1035] 5 Refused Query Refused [RFC 1035] 6 YXDomain Name Exists when it should not [RFC 2136] 7 YXRRSet RR Set Exists when it should not [RFC 2136] 8 NXRRSet RR Set that should exist does not [RFC 2136] 9 NotAuth Server Not Authoritative for zone [RFC 2136] 10 NotZone Name not contained in zone [RFC 2136]
<code>\$respsec\$</code>	Combination of all response records like answer, authority and additional records.	Answer section [record1;record2;record3] Authority section [record1;record2;record3] Addition section [record1;record2;

Keyword	Description	Example Output in logs
		record3]
<code>\$anssec\$</code>	Answer section records used.	Answer section [record1;record2; record3]
<code>\$authsec\$</code>	Authority section records used.	Authority section [record1;record2; record3]
<code>\$addsec\$</code>	Additional section records used.	Addition section [record1;record2; record3]
<code>\$src\$</code>	Packet source IP address used in the DNS logging.	20.20.20.20
<code>\$dst\$</code>	Packet destination IP address used in the DNS logging.	20.20.40.4
<code>\$spt\$</code>	Packet source port used in the DNS logging.	10012
<code>\$dpt\$</code>	Packet destination port used in the DNS logging.	53
<code>\$proto\$</code>	Packet protocol used in the DNS logging.	UDP
<code>\$snatinfo\$</code>	Source NAT information.	20.20.40.1#43014

DNS Logging Examples

DNS Request Header

The following examples show logs generated when the `request-section header` is configured:

SYSLOG Format

```
Apr 02 2019 20:47:29 Info [ACOS]:UDP 10.1.1.1.50850 30.1.10.1.53
Type=Query QueryId=28245 Opcode=Query HeaderFlag=RD|AD QDCount=1
ANCount=0 NSCount=0 ARCount=1
```

```
Apr 02 2019 20:50:28 Info      [ACOS]:TCP 10.1.1.1.52597 30.1.10.1.53
Type=Query QueryId=33027 Opcode=Query HeaderFlag=RD|AD QDCount=1
ANCCount=0 NSCount=0 ARCount=1
```

CEF Format

```
Apr 02 2019 20:48:55 Info      [ACOS]: vThunder CEF:0|A10|ADC|4.1.4-
adc-420-feat-238627|486715039831556097|Log DNS Request
Header|2|proto=UDP src=10.1.1.1 spt=39093 dst=30.1.10.1 dpt=53 cs1=Query
cs1Label=Query cn1=33511 cn1Label=Query ID cs2=Query cs2Label=Opcode
cs3=RD|AD cs3Label=Header Flag cn2=1 cn2Label=Question Count cn3=0
cn3Label=Answer Record Count cn4=0 cn4Label=Authority Record Count cn5=1
cn5Label=Additional Record Count
Apr 02 2019 20:51:43 Info      [ACOS]: vThunder CEF:0|A10|ADC|4.1.4-
adc-420-feat-238627|486715039831556097|Log DNS Request
Header|2|proto=TCP src=10.1.1.1 spt=42928 dst=30.1.10.1 dpt=53 cs1=Query
cs1Label=Query cn1=29521 cn1Label=Query ID cs2=Query cs2Label=Opcode
cs3=RD|AD cs3Label=Header Flag cn2=1 cn2Label=Question Count cn3=0
cn3Label=Answer Record Count cn4=0 cn4Label=Authority Record Count cn5=1
cn5Label=Additional Record Count
```

DNS Request Question

The following examples show logs generated when the **request-section question** is configured:

SYSLOG Format

```
Apr 02 2019 20:47:43 Info      [ACOS]:UDP 10.1.1.1.54170 30.1.10.1.53
Type=Query QueryId=7280 dhost=server.example.com QueryType=A
QueryClass=IN
Apr 02 2019 20:50:42 Info      [ACOS]:TCP 10.1.1.1.33086 30.1.10.1.53
Type=Query QueryId=37115 dhost=server.example.com QueryType=A
QueryClass=IN
```

CEF Format

```

Apr 02 2019 20:49:10 Info          [ACOS]: vThunder  CEF:0|A10|ADC|4.1.4-
adc-420-feat-238627|486715039831556098|Log DNS Request
Question|2|proto=UDP src=10.1.1.1 spt=42839 dst=30.1.10.1 dpt=53
cs1=Query cs1Label=Query cn1=32748 cn1Label=Query ID
dhost=server.example.com cs2=A cs2Label=Query Type cs3=IN cs3Label=Query
Class
Apr 02 2019 20:51:59 Info          [ACOS]: vThunder  CEF:0|A10|ADC|4.1.4-
adc-420-feat-238627|486715039831556098|Log DNS Request
Question|2|proto=TCP src=10.1.1.1 spt=38573 dst=30.1.10.1 dpt=53
cs1=Query cs1Label=Query cn1=50512 cn1Label=Query ID
dhost=server.example.com cs2=A cs2Label=Query Type cs3=IN cs3Label=Query
Class

```

DNS Request Header and Question (All)

The following examples show the logs generated when the **request-section all** is configured:

SYSLOG Format

```

Apr 02 2019 20:47:57 Info          [ACOS]:UDP 10.1.1.1.33912 30.1.10.1.53
Type=Query QueryId=62463 Opcode=Query HeaderFlag=RD|AD QDCount=1
ANCount=0 NSCount=0 ARCount=1 dhost=server.example.com QueryType=A
QueryClass=IN
Apr 02 2019 20:51:28 Info          [ACOS]:TCP 10.1.1.1.45824 30.1.10.1.53
Type=Query QueryId=27573 Opcode=Query HeaderFlag=RD|AD QDCount=1
ANCount=0 NSCount=0 ARCount=1 dhost=server.example.com QueryType=A
QueryClass=IN

```

CEF Format

```

Apr 02 2019 20:49:25 Info          [ACOS]: vThunder  CEF:0|A10|ADC|4.1.4-
adc-420-feat-238627|486715039831556099|Log DNS Request Header and
Questions|2|proto=UDP src=10.1.1.1 spt=35419 dst=30.1.10.1 dpt=53
cs1=Query cs1Label=Query cn1=6966 cn1Label=Query ID cs2=Query
cs2Label=Opcode cs3=RD|AD cs3Label=Header Flag cn2=1 cn2Label=Question
Count cn3=0 cn3Label=Answer Record Count cn4=0 cn4Label=Authority Record
Count cn5=1 cn5Label=Additional Record Count dhost=server.example.com
cs4=A cs4Label=Query Type cs5=IN cs5Label=Query Class

```



```
Apr 02 2019 20:52:14 Info [ACOS]: vThunder CEF:0|A10|ADC|4.1.4-
adc-420-feat-238627|486715039831556099|Log DNS Request Header and
Questions|2|proto=TCP src=10.1.1.1 spt=56362 dst=30.1.10.1 dpt=53
cs1=Query cs1Label=Query cn1=44728 cn1Label=Query ID cs2=Query
cs2Label=Opcode cs3=RD|AD cs3Label=Header Flag cn2=1 cn2Label=Question
Count cn3=0 cn3Label=Answer Record Count cn4=0 cn4Label=Authority Record
Count cn5=1 cn5Label=Additional Record Count dhost=server.example.com
cs4=A cs4Label=Query Type cs5=IN cs5Label=Query Class
```

DNS Response

The following examples show the logs generated when the **response-section** is configured:

SYSLOG Format

```
UDP 32.100.100.100 60 10.211.10.2 36545 Type=Response source=backend
QueryId=13984 Opcode=Query HeaderFlag=AA|RD|RA RCODE=00 QDCount=1
ANCount=3 NSCount=2 ARCount=4 response=Answer section [www.example.com
CNAME IN 86400 services.example.com;services.example.com A IN 86400
10.212.2.21;services.example.com A IN 86400 10.212.2.20;] Authority
section [example.com NS IN 86400 dns1.example.com;example.com NS IN
86400 dns2.example.com;] Addition section [dns1.example.com A IN 86400
10.212.2.20;dns1.example.com AAAA IN 86400 aaaa:bbbb::1;dns2.example.com
A IN 86400 10.212.2.22;dns2.example.com AAAA IN 86400 aaaa:bbbb::2;]
```

CEF Format

```
proto=UDP src=32.100.100.100 spt=60 dst=10.211.10.2 dpt=52050
cs1=Response cs1Label=Response cs2=backend cs2Label=Response Source
cn1=1207 cn1Label=Query ID cs3=Query cs3Label=Opcode cs4=RD|RA
cs4Label=Header Flag cs5=03 cs5Label=RCODE cn2=1 cn2Label=Question Count
cn3=0 cn3Label=Answer Record Count cn4=1 cn4Label=Authority Record Count
cn5=0 cn5Label=Additional Record Count cs6=Authority section SOA IN
10800 a.root-servers.net nstld.verisign-grs.com 2024072400 1800 900
604800 86400;] cs6Label=response content
```

DNS Custom Logging

The following examples show the logs generated when all the **custom-log** options are configured:

SYSLOG Format

Request Log

```
Nov 3 17:04:52 TH4440S-1 a10lb:[ACOS]<6> partition-id<2> <l3v> proto=UDP
src=11.11.11.10#55512 dst=11.11.11.28#53\n
```

Response Log

```
Nov 3 17:04:52 TH4440S-1 a10lb:[ACOS]<6> partition-id<2> <l3v> proto=UDP
src=11.11.11.28#53 dst=11.11.11.10#55512 snat=21.21.21.101#24152
respsrc=backend responseeeesec=Answer section [xyz.com A IN 1200
21.21.21.24;xyz.com A IN 1200 21.21.21.27;xyz.com A IN 1200
21.21.21.25;] Authority section [xyz.com NS IN 1200 ns1.xyz.com;]
Addition section [ns1.xyz.com A IN 1200 10.65.19.107;ns1.xyz.com AAAA IN
1200 2002::10:65:19:107;OPT EDNS: version: 0, flags:, udp: 4096;] \n
```

Timeout Log

```
Dec 2 15:32:40 TH4440S-1 a10lb:[ACOS]<6> partition-id<2> <l3v> Djtimeout
2002::11:11:11:9#53335 2002::11:11:11:28#53 2002::21:21:21:99#2055 UDP
B6B3 1|0|0|1 0 RD|AD xyz.com A IN 1 0 0 1\n
```

CEF Format

Request Log

```
4 Nov 3 13:24:27 TH4440S-1 CEF:0|A10|CFW|6.0.8-
d|486715327594364929|Custom DNS request log|2|proto=UDP src=11.11.11.10
spt=35178 dst=11.11.11.28 dpt=53\n
```

Response Log

```
Nov 3 13:24:27 TH4440S-1 CEF:0|A10|CFW|6.0.8-d|486715327594364930|Custom
DNS response log|2|proto=UDP src=11.11.11.28 spt=53 dst=11.11.11.10
dpt=35178 flexString1=21.21.21.101#24158 flexString1Label=SNAT Info
cs1=backend cs1Label=Response Source cs2=Answer section [xyz.com A IN
1200 21.21.21.27;xyz.com A IN 1200 21.21.21.25;xyz.com A IN 1200
21.21.21.24;] Authority section [xyz.com NS IN 1200 ns1.xyz.com;]
Addition section [ns1.xyz.com A IN 1200 10.65.19.107;ns1.xyz.com AAAA IN
1200 2002::10:65:19:107;OPT EDNS: version: 0, flags:, udp: 4096;]
cs2Label=Response Records\n
```

Timeout Log

```
Dec 2 15:39:10 TH4440S-1 CEF:0|A10|CFW|6.0.8|486715327594364931|Custom
DNS timeout log|2|src=2002::11:11:11:9 spt=40055 dst=2002::11:11:11:28
dpt=53 flexString1=2002::21:21:21:99#2065 flexString1Label=SNAT Info
proto=UDP cs1=5B54 cs1Label=Query ID cs2=1|0|0|1 cs2Label=qd|an|ns|ar
cs3=0 cs3Label=Opcode cs4=RD|AD cs4Label=Header Flag cs5=xyz.com A IN
cs5Label=Query Info cn1=1 cn1Label=QDCOUNT cn2=0 cn2Label=ANCOUNT cn3=0
cn3Label=NSCOUNT cn4=1 cn4Label=ARCOUNT\n
```

RPZ Filtering

The following examples show sample custom log entries generated when RPZ filtering is applied:

SYSLOG Example

Request Log

```
Dec 4 16:13:57 ACOS a10lb:[ACOS]<6> rcode=NoError arcount=1 ancount=0
qdcount=1 recordcounters=1|0|0|1 queryclass=IN src=1.1.0.206
dst=1.1.0.211 dpt=53 proto=UDP queryname=localdata88.com querytype=A\n
```

Request Log

```
Dec 4 16:13:57 ACOS a10lb:[ACOS]<6> responsesrc=rpz\n
```

CEF Example

Request Log

```
Dec 4 15:54:09 ACOS CEF:0|A10|CFW|6.0.8|486715327594364929|Custom DNS
request log|2|cs1=NoError cs1Label=Rcode Name cn1=1 cn1Label=ARCOUNT
cn2=0 cn2Label=ANCOUNT cn3=1 cn3Label=QDCOUNT cs2=1|0|0|1
cs2Label=qd|an|ns|ar cs3=IN cs3Label=Query Class src=1.1.0.206
dst=1.1.0.211 spt=44962 dpt=53 proto=UDP dhost=localdata88.com cs4=A
cs4Label=Query Type\n
```

Response Log

```
Dec 4 15:54:09 ACOS CEF:0|A10|CFW|6.0.8|486715327594364930|Custom DNS
response log|2|cs1=rpz cs1Label=Response Source\n
```

The following examples show sample standard log entries generated when RPZ filtering is applied:

SYSLOG Example

Request Log

```
Dec 4 16:13:57 ACOS a10lb:[ACOS]<6> UDP 1.1.0.206 55196 1.1.0.211 53
Type=Query QueryId=16008 Opcode=Query HeaderFlag=RD|AD QDCount=1
ANCount=0 NSCount=0 ARCount=1 dhost=localdata88.com QueryType=A
QueryClass=IN\n
```

Response Log

```
Dec 4 16:13:57 ACOS a10lb:[ACOS]<6> UDP 1.1.0.211 53 1.1.0.206 55196
Type=Response source=rpz QueryId=16008 Opcode=Query HeaderFlag=RD|AD
RCODE=00 QDCount=1 ANCount=1 NSCount=0 ARCount=0 response=Answer section
[localdata88.com A IN 3600 1.1.1.10;] \n
```

CEF Example

Request Log

```
Dec 4 15:54:09 ACOS CEF:0|A10|CFW|6.0.8|486715314709463043|Log DNS
Request Header and Questions|2|proto=UDP src=1.1.0.206 spt=44962
dst=1.1.0.211 dpt=53 cs1=Query cs1Label=Query cn1=34652 cn1Label=Query
ID cs2=Query cs2Label=Opcode cs3=RD|AD cs3Label=Header Flag cn2=1
cn2Label=Question Count cn3=0 cn3Label=Answer Record Count cn4=0
cn4Label=Authority Record Count cn5=1 cn5Label=Additional Record Count
dhost=localdata88.com cs4=A cs4Label=Query Type cs5=IN cs5Label=Query
Class\n
```

Response Log

```
Dec 4 15:54:09 ACOS CEF:0|A10|CFW|6.0.8|486715314709463052|Log ipv4 DNS
Response Header and Answer with Rcode|2|proto=UDP src=1.1.0.211 spt=53
dst=1.1.0.206 dpt=44962 cs1=Response cs1Label=Response cs2=rpz
cs2Label=Response Source cn1=34652 cn1Label=Query ID cs3=Query
cs3Label=Opcode cs4=RD|AD cs4Label=Header Flag cs5=00 cs5Label=RCODE
cn2=1 cn2Label=Question Count cn3=1 cn3Label=Answer Record Count cn4=0
cn4Label=Authority Record Count cn5=0 cn5Label=Additional Record Count
cs6=Answer section [localdata88.com A IN 3600 1.1.1.10;]
cs6Label=response content\n
```

CLI Configuration

To configure RDATA type in the DNS logging, use the following commands:

```
ACOS(config)#slb template dns-logging dnslog_template_name
ACOS(config-dns-logging)#response-type config
ACOS(config-dns-logging-response-type)#type TXT
```

To configure RCODE and response section in the DNS logging, use the following commands:

```
ACOS(config)#slb template dns-logging dnslog_template_name
ACOS(config-dns-logging)# response-include-rcode
ACOS(config-dns-logging)# response-section all
```

To configure custom request log in the DNS logging template, use the following commands:

```
ACOS(config)#slb template dns-logging log_template_name
ACOS(config-dns-logging)# custom-log request
ACOS(config-dns-logging-custom-log:request)# format "queryid is $queryId$
opc is $opcode$ rcode is $rcode$ rcode string is $rcodestr$"
ACOS(config-dns-logging-custom-log:request)# enable
ACOS(config-dns-logging-custom-log:request)# exit
```

To configure custom response log in the DNS logging template, use the following commands:

```
ACOS(config-dns-logging)# custom-log response
ACOS(config-dns-logging-custom-log:response)# format "queryid is $queryId$
opc is $opcode$ Response src is $respsrc$ Rcode is $rcode$ NAT info is
$snatinfo$"
ACOS(config-dns-logging-custom-log:response)# enable
ACOS(config-dns-logging-custom-log:response)# exit
```

To configure custom timeout log, use the following commands:

```
ACOS(config)# slb template dns-logging log_template_name
ACOS(config-dns-logging)# custom-log timeout
ACOS(config-dns-logging-custom-log:timeout)# format "Djtimeout $src##$spt$
$dst##$dpt$ $snatinfo$ $proto$ $queryid$ $recordcounters$ $opcode$
$headerflag$ $queryinfo$ $qdcoun$ $ancoun$ $nscount$ $arcount$"
ACOS(config-dns-logging-custom-log:timeout)# enable
```

To enable standard RPZ-based logging under a DNS logging template, use the following commands:

```
ACOS(config)#slb template dns-logging log_template_name
ACOS(config-dns-logging)# standard-log request
ACOS(config-dns-logging-standard-log:request)# log-filter RPZ
```

To enable custom RPZ-based logging under a DNS logging template, use the following commands:

```
ACOS(config)#slb template dns-logging log_template_name
ACOS(config-dns-logging)# custom-log request
ACOS(config-dns-logging-custom-log:request)# log-filter RPZ
```

GUI Configuration

1. Navigate to **ADC >> Templates >> L7 Protocols >> Create >> DNS Logging**.

The following options are displayed on **Create DNS Logging Template**:

- **Name** - Specify the name of the DNS Logging template.
- **Disable DNS-Logging Template** - Select the option to disable the DNS Logging template.
- **Protocol** - Select the DNS protocol from **TCP**, **UDP** or **Both**.

- **DNS-Logging Template Type** - Select the template type from **Query**, **Response** or **Both**.
 - **Request Section** - Select one of the options from **Header**, **Question**, **All**.
 - **Query Type Filter** - Select one of the options from **Header**, **Answer**, **All**.
2. Navigate to **ADC >> Templates >> L7 Protocols >> Create >> DNS**.

The following option is added:

- **DNS-Logging Template** - Select the DNS Logging template from the list.

NOTE: If the option **Disable DNS-Logging Template** is selected on the page **Create DNS Logging Template**, the DNS Logging template does not appear in the drop-down list.

DNS Queries per Second Logging

DNS Queries Per Second (QPS) measure the rate at which queries are received within one second. ACOS provides real-time notifications for DNS QPS logging by calculating the DNS QPS value per virtual port (vport) in an SLB DNS template. High and low QPS thresholds are configured and enable event logging and SNMP traps. When the QPS value exceeds configured high threshold, drops below the configured low threshold, or returns to a normal level, the system generates event logs and SNMP traps based on the current QPS value.

CLI Configuration

To configure DNS QPS logging, perform the following steps:

1. Configure the SLB server.

```
ACOS(config)# slb server sv 10.10.10.1
ACOS(config-rserver)# port 53 tcp
ACOS(config-rserver)# port 53 udp
ACOS(config-rserver)# exit
```

2. Configure the SLB service group for UDP and TCP.

```
ACOS(config)# slb service-group svg_udp udp
ACOS(config-slb svc group)# member sv 53
ACOS(config-slb svc group)# exit
ACOS(config)# slb service-group svg_tcp tcp
ACOS(config-slb svc group)# member sv 53
ACOS(config-slb svc group)# exit
```

3. Define the SLB template and enable query rate monitoring with specified thresholds and event logging.

```
ACOS(config)# slb template dns temp_dns
ACOS(config-dns)# query-rate-monitoring threshold-log high 20 low 5
enable
ACOS(config-dns)# exit
```

In the above command:

- Ensure the high threshold value is greater than the low threshold value.
- Configure the high and low threshold values to calculate only the current QPS.
- Use the **enable** option with the high and low threshold values to enable the generation of the event-logs whenever QPS value crosses the configured threshold.

4. Configure the SLB virtual server.

```
ACOS(config)# slb virtual-server vs 10.10.10.99
ACOS(config-slb vserver)# port 53 dns-tcp
ACOS(config-slb vserver-vport)# service-group svg_tcp
ACOS(config-slb vserver-vport)# template dns temp_dns
ACOS(config-slb vserver-vport)# exit
ACOS(config-slb vserver)# port 53 dns-udp
ACOS(config-slb vserver-vport)# service-group svg_udp
ACOS(config-slb vserver-vport)# template dns temp_dns
ACOS(config-slb vserver-vport)# exit
```

5. Enable SNMP service on the load balancer.


```
ACOS(config)# snmp-server enable service
ACOS(config)# snmp-server host 192.168.3.67 version v1 public udp-
port 162
ACOS(config)# snmp-server host 192.168.3.67 version v2c user1
```

6. Enable SNMP trap for DNS QPS logging.

```
ACOS(config)# snmp-server enable traps slb vip-port-qps
```

The SNMP traps are triggered during threshold events only if the virtual port is associated with a DNS template and the high/low values for QPS monitoring in that template are configured.

Show command

To view the DNS QPS logs rate, use the following show command:

```
ACOS(config)#show log
Log Buffer: 30000
Mar 20 2024 08:13:43 Warning [ACOS]:Virtual server vs service port 53
type dns-udp QPS 3 is below low threshold 5
```

This syslog indicates that the QPS for the specified SLB virtual server's service port has dropped below the configured low threshold.

Event Logs

The following event logs are available for DNS QPS logging:

Event Log	Description
Below low threshold	The QPS value goes below the configured low threshold value.
Back to normal level	The QPS value returns to normal value from high or low threshold value.
Above high threshold	The QPS value exceeds the configured high threshold value.

SNMP OID

The following SNMP OIDs are assigned to each DNS QPS logging event type:

Event Type	Description	SNMP OID
Normal to high event	Transition from normal QPS level to high QPS level.	1.3.6.1.4.1.22610.2.4.3.12.2.6.13
High to normal event	Transition from high QPS level to normal QPS level.	1.3.6.1.4.1.22610.2.4.3.12.2.6.14
Normal to low event	Transition from normal QPS level to low QPS level.	1.3.6.1.4.1.22610.2.4.3.12.2.6.15
Low to normal event	Transition from low QPS level to normal QPS level.	1.3.6.1.4.1.22610.2.4.3.12.2.6.16

Limitations

- Low-latency mode is not supported.

Authenticate DNS Requests

ACOS can authenticate DNS requests received over UDP by directing the client to re-send the DNS request over TCP. By default, DNS requests from clients are not authenticated.

When authentication of DNS-over-UDP is enabled, ACOS drops DNS requests from the client if they are sent over UDP and sends the client a DNS Truncate message. This action essentially informs the client that it must re-send the DNS request over TCP in order to pass the DNS authentication.

Additionally, the DNS cache sharing for TCP and UDP ability is also provided. This ability is to cache TCP based DNS queries and is interchangeable with queries made to UDP and TCP based VIPs. This feature is implemented in tandem with caching so that the initial request is not only redirected to TCP, but is then cached so that a second request is not made to the DNS server.

CLI Configuration

- To enable authenticating client requests by directing the client to retry the connection over TCP (instead of UDP), use the following command:

```
ACOS(config)#slb template dns dns_template
ACOS(config-dns)#redirect-to-tcp-port
```

- To enable DNS Cache Sharing, use the following command:

```
ACOS(config)#slb template dns dns_template
ACOS(config-dns)#enable-cache-sharing
```

Configuration Examples – DNS Optimization

The following subsections show how to configure the types of DNS caching policies listed at the beginning of this section. Most of these examples use LIDs configured within the DNS template, instead of GLIDs. For an example that uses a GLID, see [Rate-Based DNS Caching with Rate Limiting \(CLI Example\)](#).

Control Caching on Per-VIP Basis (CLI Example)

To implement this type of DNS caching policy:

- Configure a DNS caching policy in which the default action is to cache, and bind it to the DNS virtual port for which you want to cache DNS replies.
- Configure another DNS caching policy, in which the default action is not to cache, and bind it to the DNS virtual port for which you do not want to cache DNS replies.

In this example, default settings are used for all DNS caching parameters except the default cache action.

The following commands configure the DNS caching templates:

```
ACOS(config)# slb template dns dns-enable-template
ACOS(config-dns)# default-policy cache
ACOS(config-dns)# exit
ACOS(config)# slb template dns dns-disable-template
ACOS(config-dns)# default-policy nocache
ACOS(config-dns)# exit
```

The following commands configure the real server and service group:

```
ACOS(config)# slb server dns-opt 10.10.10.88
ACOS(config-real server)# port 53 udp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
ACOS(config)# slb service-group group53 udp
ACOS(config-slb svc group)# member dns-opt 53
```

```
ACOS(config-slb svc group-member:53)# exit
ACOS(config-slb svc group)# exit
```

The following commands bind the dns-enable-template to a VIP:

```
ACOS(config)# slb virtual-server vip-cache 192.168.10.10
ACOS(config-slb vserver)# port 53 dns-udp
ACOS(config-slb vserver-vport)# service-group group53
ACOS(config-slb vserver-vport)# template dns dns-enable-template
ACOS(config-slb vserver-vport)# exit
ACOS(config-slb vserver)# exit
```

The following commands bind the dns-disable-template to a different VIP:

```
ACOS(config)# slb virtual-server vip-nocache 192.168.10.11
ACOS(config-slb vserver)# port 53 dns-udp
ACOS(config-slb vserver-vport)# service-group group53
ACOS(config-slb vserver-vport)# template dns dns-disable-template
ACOS(config-slb vserver-vport)# exit
ACOS(config-slb vserver)# exit
```

Since the default action is nocache, dns-disable-template is not needed for vip-nocache.

Control Caching on Per-Record Basis (CLI Example)

To implement this type of DNS caching policy:

- Create a class list that contains match rules for domain names to perform DNS caching. Associate each domain name string with an LID. In DNS templates, LIDs contain specific caching policies. In this example:
 - Allow caching of all entries for “example.com”; for example, “mail.example.com”, “ftp.example.com”, and so on.
 - Deny caching of entries for “www.example.com”.
- Configure a DNS template with default action nocache, add the class list to the template, and configure these LIDs:
 - LID 1 – DNS cache-enable
 - LID 2 – DNS cache-disable
- Bind the DNS template to the DNS virtual port.

The following commands configure the class list:

```
ACOS(config)# class-list dns-record
ACOS(config-class list)# dns ends-with example.com lid 1
ACOS(config-class list)# dns contains www.example.com lid 2
ACOS(config-class list)# exit
```

The following commands configure the DNS template:

```
ACOS(config)# slb template dns dns-template
ACOS(config-dns)# default-policy nocache
ACOS(config-dns)# class-list dns-record
ACOS(config-dns-class-list:dns-record)# lid 1
ACOS(config-dns-class-list:dns-record-li...)# dns cache-enable
ACOS(config-dns-class-list:dns-record-li...)# exit
ACOS(config-dns-class-list:dns-record)# lid 2
ACOS(config-dns-class-list:dns-record-li...)# dns cache-disable
ACOS(config-dns-class-list:dns-record-li...)# exit
ACOS(config-dns-class-list:dns-record)# exit
ACOS(config-dns)# exit
```

The following commands configure the real server and service group:

```
ACOS(config)# slb server dns-opt 10.10.10.88
ACOS(config-real server)# port 53 udp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
ACOS(config)# slb service-group group53 udp
ACOS(config-slb svc group)# member dns-opt 53
ACOS(config-slb svc group-member:53)# exit
ACOS(config-slb svc group)# exit
```

The following commands bind the DNS template to the DNS virtual port:

```
ACOS(config)# slb virtual-server vip-cache 192.168.10.10
ACOS(config-slb vserver)# port 53 dns-udp
ACOS(config-slb vserver-vport)# service-group group53
ACOS(config-slb vserver-vport)# template dns dns-template
ACOS(config-slb vserver-vport)# exit
ACOS(config-slb vserver)# exit
```

Rate-Based DNS Caching with Rate Limiting (CLI Example)

To implement this type of DNS caching policy:

- Create a class list that contains match rules for the domain names for which to perform DNS caching. Associate the URL string with an LID.
- Configure a GLID that enables caching when a specified connection rate is reached.
- Configure a DNS template that specifies the default action (in this example, nocache).
- Bind the DNS template to the DNS virtual port.

Although this example uses a GLID, you can use a LID within the DNS template instead.

The following commands configure the class list:

```
ACOS(config)# class-list dns-record
ACOS(config-class list)# dns contains .example2.com glid 1
ACOS(config-class list)# exit
```

The following commands configure the GLID:

```
ACOS(config)# glid 1
ACOS(config-global lid)# dns cache-disable
ACOS(config-global lid)# conn-rate-limit 1000 per 10
ACOS(config-global lid)# over-limit-action dns-cache-enable
ACOS(config-global lid)# exit
```

The following commands configure the DNS template:

```
ACOS(config)# slb template dns dns-template
ACOS(config-dns)# default-policy nocache
ACOS(config-dns)# class-list dns-record
ACOS(config-dns-class-list:dns-record)# lid 1
ACOS(config-dns-class-list:dns-record-li...)# exit
ACOS(config-dns-class-list:dns-record)# exit
ACOS(config-dns)# exit
```

The following commands configure the real server and service group:

```
ACOS(config)# slb server dns-opt 10.10.10.88
ACOS(config-real server)# port 53 udp
ACOS(config-real server-node port)# exit
```

```
ACOS(config-real server)# exit
ACOS(config)# slb service-group group53 udp
ACOS(config-slb svc group)# member dns-opt 53
ACOS(config-slb svc group-member:53)# exit
ACOS(config-slb svc group)# exit
```

The following commands bind the DNS template to the DNS virtual port:

```
ACOS(config)# slb virtual-server vip-cache 192.168.10.10
ACOS(config-slb vserver)# port 53 dns-udp
ACOS(config-slb vserver-vport)# service-group group53
ACOS(config-slb vserver-vport)# template dns dns-template
ACOS(config-slb vserver-vport)# exit
ACOS(config-slb vserver)# exit
```

DNS Record Weighting for Selective Cache Entry Aging (CLI Example)

To implement this type of DNS caching policy:

- Create a class list that contains match rules for the domain names for which to perform DNS caching. Associate each URL string with an LID.
- Configure a DNS template with default action nocache, add the class list to the template, and configure these LIDs:
 - LID 1 – DNS cache-enable, weight 1
 - LID 2 – DNS cache-enable, weight 2
- Bind the DNS template to the DNS virtual port.

The following commands configure the class list:

```
ACOS(config)# class-list dns-record
ACOS(config-class list)# dns contains example-corp1.com lid 1
ACOS(config-class list)# dns contains example-corp2.com lid 2
ACOS(config-class list)# exit
```

The following commands configure the DNS template:

```
ACOS(config)# slb template dns dns-template
ACOS(config-dns)# default-policy nocache
ACOS(config-dns)# class-list dns-record
ACOS(config-dns-class-list:dns-record)# lid 1
ACOS(config-dns-class-list:dns-record-li...)# dns cache-enable
```

```
ACOS(config-dns-class-list:dns-record-li...)# dns weight 1
ACOS(config-dns-class-list:dns-record-li...)# exit
ACOS(config-dns-class-list:dns-record)# lid 2
ACOS(config-dns-class-list:dns-record-li...)# dns cache-enable
ACOS(config-dns-class-list:dns-record-li...)# dns weight 2
ACOS(config-dns-class-list:dns-record-li...)# exit
ACOS(config-dns-class-list:dns-record)# exit
ACOS(config-dns)# exit
```

The following commands configure the real server and service group:

```
ACOS(config)# slb server dns-opt 10.10.10.88
ACOS(config-real server)# port 53 udp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
ACOS(config)# slb service-group group53 udp
ACOS(config-slb svc group)# member dns-opt 53
ACOS(config-slb svc group-member:53)# exit
ACOS(config-slb svc group)# exit
```

The following commands bind the DNS template to the DNS virtual port:

```
ACOS(config)# slb virtual-server vip-cache 192.168.10.10
ACOS(config-slb vserver)# port 53 dns-udp
ACOS(config-slb vserver-vport)# service-group group53
ACOS(config-slb vserver-vport)# template dns dns-template
ACOS(config-slb vserver-vport)# exit
ACOS(config-slb vserver)# exit
```

Throttling Based on Domain Name (CLI Example)

To implement this type of DNS caching policy:

- Create a class list with match rules for domain names to throttle. Associate each URL string with an LID.
- Assign a very low connection or request rate (for example, 1).
- Optionally, enable lockout.

The following commands configure the class list:

```
ACOS(config)# class-list dns-throttle-cl
ACOS(config-class list)# dns contains example.com lid 1
```



```
ACOS(config-class list)# exit
```

The following commands configure the DNS template:

```
ACOS(config)# slb template dns dns-throttle
ACOS(config-dns)# class-list dns-throttle-cl
ACOS(config-dns-class-list:dns-throttle-cl)# lid 1
ACOS(config-dns-class-list:dns-throttle-c...)# conn-rate-limit 1 per 65535
ACOS(config-dns-class-list:dns-throttle-c...)# over-limit-action lockout
1023
ACOS(config-dns-class-list:dns-throttle-c...)# exit
ACOS(config-dns-class-list:dns-throttle-cl)# exit
ACOS(config-dns)# exit
```

The following commands configure the real server and service group:

```
ACOS(config)# slb server dns-opt 10.10.10.88
ACOS(config-real server)# port 53 udp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
ACOS(config)# slb service-group group53 udp
ACOS(config-slb svc group)# member dns-opt 53
ACOS(config-slb svc group-member:53)# exit
ACOS(config-slb svc group)# exit
```

The following commands bind the DNS template to the DNS virtual port:

```
ACOS(config)# slb virtual-server vip-cache 192.168.10.10
ACOS(config-slb vserver)# port 53 dns-udp
ACOS(config-slb vserver-vport)# service-group group53
ACOS(config-slb vserver-vport)# template dns dns-throttle
ACOS(config-slb vserver-vport)# exit
ACOS(config-slb vserver)# exit
```

Logging for DNS Caching (CLI Example)

This example enables logging for DNS caching.

```
ACOS(config)# slb template dns dns-template
ACOS(config-dns)# dns-log-enable period 300
ACOS(config-dns)# exit
```

Logging for DNS caching will be enabled for each virtual DNS port that uses this template. Logs will be generated every 5 minutes.

Displaying DNS Caching Information (CLI Example)

The following commands show DNS caching information.

```
ACOS# show dns cache entry
```

T = Type, C = Class, W = weight								
Qlen = Query length, Rlen = Response length								
Domain	T	C	Qlen	Rlen	TTL	Age	W	
Hit								
-----+-----+-----+-----+-----+-----+-----+-----+-----								
ad.example.com	1	1	19	127	666	300	2	
100								
example.com	1	1	16	68	300	240	1	
0								
www.examplelive.com	1	1	24	142	666	360	2	
257								
.example2.com	1	1	16	89	666	420	2	
0								
example-corp1.com	1	1	21	74	666	420	2	
11								
example-corp2.com	1	1	17	104	666	420	2	
12								

NOTE:

- The T (Type) column lists the DNS record type. For a list, see the following: http://en.wikipedia.org/wiki/List_of_DNS_record_types
- If logging is enabled, the Hit counter is reset after each log entry is created. (See [DNS Cache Logging](#) and [Logging for DNS Caching \(CLI Example\)](#).)
- The counter in the Age column is always a multiple of 60. This is because the age is rounded up to the next 60-second cache refresh interval. ACOS refreshes the cache every 60 seconds. An entry that has aged out is removed at some point during the 60-second cache refresh.

```
ACOS# show dns cache client
Client class list statistics:
          F = Flag (C-Connection, R-Request), Over-RL = Over rate
limit
Domain          Destination          F Rate      Over-RL      lock
      lock-time
-----+-----+-----+-----+-----+-----+-----
-----+-----
ad.example.com   192.22.35.90:53      C 0          0           0
      0
example.com      192.22.35.90:53      C 0          0           0
      0
www.examplelive.com 192.22.35.90:53      C 0          30          0
      0
.example2.com    192.22.35.90:53      C 0          0           0
      0
example-corp1.com 192.22.35.90:53      C 0          0           0
      0
example-corp2.com 192.22.35.90:53      C 0          0           0
      0

ACOS# show dns cache statistics
Total allocated: 266
Total freed: 96
Total query: 1720
Total server response: 485
```

```

Total cache hit: 1218
Query not passed: 0
Response not passed: 0
Response answer not passed: 0
Query encoded: 0
Response encoded: 0
Query with multiple questions: 0
Response with multiple questions: 0
Response with multiple answers: 0
Response with short TTL: 0
Total aged out: 97
Total aged for lower weight: 0
Total stats log sent: 69
Current allocate: 170
Current data allocate: 23

```

Authentication for DNS Requests (CLI Example)

This example creates the DNS template “temp_dns1” and applies it to two virtual ports so enable **enable-cache-sharing** and **redirect-to-tcp** port options. This example uses these components without providing their configuration: **wild** (class list), **sg-dns** (service group), and **svg_v6_0** (service group).

```

ACOS(config)# slb template dns temp_dns1
ACOS(config-dns)# max-cache-size 60000
ACOS(config-dns)# max-cache-entry-size 4096
ACOS(config-dns)# dns-log-enable period 1
ACOS(config-dns)# enable-cache-sharing
ACOS(config-dns)# class-list wild
ACOS(config-dns-class-list:wild)# lid 1
ACOS(config-dns-class-list:wild-lid:1)# dns cache-enable
ACOS(config-dns-class-list:wild-lid:1)# dns weight 7
ACOS(config-dns-class-list:wild-lid:1)# dns ttl 65535
ACOS(config-dns-class-list:wild-lid:1)# exit
ACOS(config-dns-class-list:wild)# exit
ACOS(config-dns)# exit
ACOS(config)# slb virtual-server vs1 10.105.1.111
ACOS(config-slb vserver)# port 53 dns-udp
ACOS(config-slb vserver-vport)# service-group sg-dns
ACOS(config-slb vserver-vport)# template dns temp_dns1
ACOS(config-slb vserver-vport)# exit

```

```
ACOS(config-slb vserver) # port 53 dns-tcp
ACOS(config-slb vserver-vport) # service-group svg_v6_0
ACOS(config-slb vserver-vport) # template dns temp_dns1
ACOS(config-slb vserver-vport) # exit
ACOS(config-slb vserver) # port 5353 dns-udp
ACOS(config-slb vserver-vport) # service-group sg-dns
ACOS(config-slb vserver-vport) # template dns temp_dns1
ACOS(config-slb vserver-vport) # exit
ACOS(config-slb vserver) # port 5353 dns-tcp
ACOS(config-slb vserver-vport) # service-group svg_v6_0
ACOS(config-slb vserver-vport) # template dns temp_dns1
ACOS(config-slb vserver-vport) # exit
ACOS(config-slb vserver) # exit
```

In the example above, the DNS template “temp_dns1” is applied to two virtual ports, (53 and 5353). With **enable-cache-sharing** enabled, those ports will share the DNS cache, which will save memory costs.

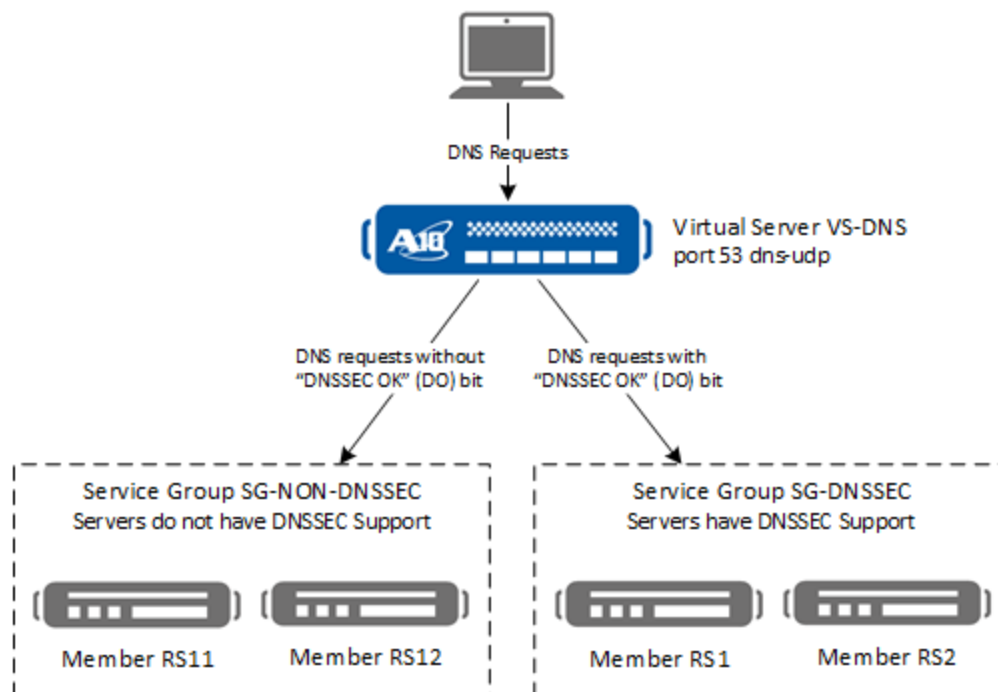
The following example shows you how to disable cache sharing in your configured DNS template:

```
ACOS(config) # slb template dns temp_dns1
ACOS(config-dns) # no enable-cache-sharing
ACOS(config-dns) # exit
```

Load Balancing with the “DNSSEC OK” (DO) Bit

This feature enables the ACOS device to load balance DNS requests from clients supporting DNS Security Extensions (DNSSEC) to servers supporting the same ([Figure 63](#)). This feature is also supported using aFlex scripts (`DNS::header`).

Figure 63 : Load Balancing with the “DNSSEC OK” (DO) Bit



The device checks the header of the UDP or TCP packet for an OPT field, and checks the OPT field for the “DNSSEC OK” or DO bit. If not found, the DNS request is load-balanced to a service group for DNS servers not supporting DNSSEC. If found, the request is sent to a service group for servers providing DNSSEC support.

Configuring Load Balancing with the DNSSEC OK Bit (CLI Example)

This configuration implements the topology in [Figure 63](#). This example uses UDP, but TCP can also be used.

Configure the SG-DNSSEC service group for servers supporting DNSSEC:

```

ACOS(config)$ slb server RS1 10.0.0.1
ACOS(config-real server)$ exit
ACOS(config)$ slb server RS2 10.0.0.2
ACOS(config-real server)$ exit
ACOS(config)$ slb service-group SG-DNSSEC udp
ACOS(config-slb svc group)$ member RS1 53001
ACOS(config-slb svc group-member:53001)$ exit
ACOS(config-slb svc group)$ member RS2 53001
ACOS(config-slb svc group-member:53001)$ exit
  
```

Configure the SG-NON-DNSSEC service group for servers not supporting DNSSEC:

```
ACOS(config)$ slb server RS11 10.0.0.11
ACOS(config-real server)$ exit
ACOS(config)$ slb server RS12 10.0.0.12
ACOS(config-real server)$ exit
ACOS(config)$ slb service-group SG-NON-DNSSEC udp
ACOS(config-slb svc group)$ member RS11 53001
ACOS(config-slb svc group-member:53001)$ exit
ACOS(config-slb svc group)$ member RS12 53001
ACOS(config-slb svc group-member:53001)$ exit
ACOS(config-slb svc group)$ exit
```

Create the DNS SLB template TMP-DNSSEC and apply it to the SG-DNSSEC service group.

```
ACOS(config)$ slb template dns TMP-DNSSEC
ACOS(config-dns)$ dnssec-service-group SG-DNSSEC
ACOS(config-dns)$ exit
```

Configure the virtual server and port on the ACOS device, and associate them with the proper service group and template.

```
ACOS(config)$ slb virtual-server VS-DNS 10.10.10.10
ACOS(config-slb vserver)$ port 53 dns-udp
ACOS(config-slb vserver-vport)$ service-group SG-NON-DNSSEC
ACOS(config-slb vserver-vport)$ template dns TMP-DNSSEC
ACOS(config-slb vserver-vport)$ exit
ACOS(config-slb vserver)$ exit
```

View the new Layer 4 statistics counter related to this feature:

```
ACOS(config)$ show slb l4
```

The “DNSSEC switch” field displays the number of DNSSEC packets switch to the service group supporting DNSSEC.

DNS Recursive Resolver

When the DNS recursive feature is enabled, ACOS forwards the queries to the root DNS servers by default, and performs the recursive lookup in the scenarios mentioned below:

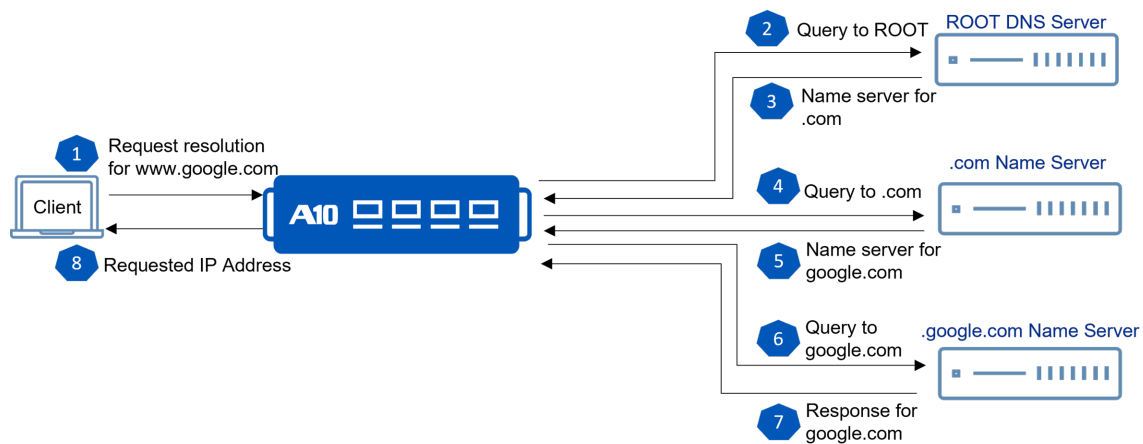
- If no service-group is bound under the virtual port where this feature is deployed.
- If the servers in the service-group bound to the virtual port are DOWN (due to health check) or DISABLED.
- If the DNS server in the service-group bound to the virtual port responds to the client's query without a matching record.

When the root DNS server has no final answer, it responds with an NS (nameserver) delegation (referral) response. ACOS then initiates a query to the server pointed to in the referral response. This process is repeated until the final response is fetched and delivered to the client.

NOTE:

- When the DNS template is bound to the virtual port, the virtual port is marked 'functional up', even if no servers are up due to a health check in the service-group.
- If the `dns-cache` feature is enabled, final response and the response of nameserver (NS) address resolving will be cached.

Figure 64 : Workflow of ACOS with recursive resolution



Supported Features

- Supports dns-udp and dns-tcp type virtual server ports only.
- Supports DNS security features with recursive DNS resolver under the SLB template DNS.

Limitation

- Does not support stateful query-ID-based load balancing (`query-id-switch`) when enabled on any DNS template in the same L3v partition.
- The DNS recursive resolver does not completely support the resolution of DNS queries of the type=ANY.

DNS Recursive Resolver Parameters

This section describes the various configurable parameters for DNS recursive resolver.

For more information, see *Command Line Interface Reference*.

- `ipv4-nat-pool/ipv6-nat-pool` configures the source NAT pool, which is the mandatory configuration.

IP protocol translation is supported. For example, DNS VIP is IPv4, and recursive DNS lookup IP is IPv6.

- `default-recursive` By default, ACOS performs recursive resolution to serve the client queries when this option is enabled. This feature is applicable on a per DNS virtual port (dns-tcp/dns-udp).

ACOS will forward the query to the service-group only when either condition is met.

- The service-group forwarding features (such as aflex, PBSLB, or DNS template dnssec-service-group) are configured in the virtual port.
- The query name matches the `hostnames`, if configured.

- **dnssec-validation**, if enabled, ACOS recursive resolver performs DNSSEC validation on responses received before the resolver continues to resolve a CNAME or sends the response to clients.
- **hostnames** bind a DNS type class list to perform TLD resolution for a subset of domain names instead of all the DNS queries. If the query name match then ACOS performs recursive resolution.

However, if **default-recursive** is configured, the **hostnames** behavior changes as described below:

- When the query name matches, the requests are forwarded to the bound service-group.
- When the query name does not match, the requests are forwarded to the recursive resolver.
- **dns-cache-enable** enables caching the responses only when configured globally under 'slb common' configuration mode.
 - When a new query is received from the client, ACOS first searches the dns-cache for a DNS response matching the client's FQDN, query type, and query class.
 - If not found, NS records are searched in the cache before starting the recursive resolution process.

The NS record search is done to find the authoritative name server closest to the client's requested FQDN to start the recursive resolution process instead of approaching the default root DNS servers.

- The closest NS match from the cache is then used only if a corresponding valid A/AAAA record exists.

If ACOS does not find a corresponding A/AAAA record in the cache for the closest NS match, a full resolution process is used (the recursive resolution starts from the root DNS servers).

- **edns-udp-size** - Specifies the UDP payload size to advertise in the EDNS(0) OPT record for recursive resolution DNS queries generated by ACOS.
 - Applicable to **dns-udp** and **dns-tcp** virtual port.
 - Range: 512–4096 bytes
 - Default: 4096 bytes

- The configured **edns-udp-size** value must be greater than or equal to the **max-udp-size** value configured under that DNS template. If no **max-udp-size** value is configured under the DNS template, then **edns-udp-size** value must be greater than or equal to **dns-max-udp-size** value configured globally.
- **parallel-queries** sets the number of parallel queries that can be sent to the servers. This command improves the efficiency of recursive resolution as it sends concurrent queries to multiple resolvers and considers the fastest response. It supports both, the TCP and UDP protocols.
- **ns-cache-lookup** is the NS records extracted from the authority section, and the A/AAAA glue records that are extracted from the additional section in the DNS (referral) responses are cached in addition to the final response. This operation accelerates the resolution process.
 - ACOS caches only the final answer that corresponds with **www.google.com**.
 - If the client now requests **mail.google.com**, the whole process is repeated, starting from approaching the root DNS servers.
 - ACOS caches the NS records for the **.com** server as well as **google.com** and any associated A/AAAA glue records extracted from the referral responses.
 - The resolution for **mail.google.com** can start directly with the NS of **google.com** instead of the root DNS servers.
- **full-response** adds the 'Authority' and 'Additional' sections in the response.

The minimal response is the default behavior. Therefore, the 'Authority' and 'Additional' sections are removed from the response before returning the output to the client.

- The 'Authority' and 'Additional' sections are not removed for EDNS(0) as it follows bind server behavior.
- Complete response is saved in the cache. If the cache is shared, other users can access the response.
- **force-cname-resolution** enables or disables the CNAME resolution option when the matching answer record exists in the CNAME response packet.
- **max-trials** configures the maximum number of total trials ACOS performs, cumulative of all levels based on per client DNS query, before quitting the

recursive process. When the default value (0) is used, the total number of trials attempted by ACOS to resolve a client's query is not limited. Still, the `retries-per-level` option mentioned below limits the number of queries at each level.

- `request-for-pending-resolution` configures a specific action to be taken for such queries. If ACOS receives client queries for an FQDN and query type for which recursive resolution is already initiated.
 - `drop` - drops the requests.
 - `respond-with-servfail` - Responds with a SERVFAIL DNS response (Default).
 - `start-new-resolution` - Starts new/additional recursive resolution requests.
- `retries-per-level` configures the maximum number of retries at each level of the DNS resolution process.
- `udp-retry-interval` configures the UDP DNS retry interval value.
- `use-service-group-response` enables or disables the backend service group response and skips the recursive resolver.

CLI Configuration

1. Configures a recursive DNS resolver.

```
ACOS(config)# slb template dns dns1
ACOS(config-dns)# recursive-dns-resolution
ACOS(config-dns-recursive-dns-resolution)# host1
ACOS(config-dns-recursive-dns-resolution)# ipv4-nat-pool eth2ipv4_1
ACOS(config-dns-recursive-dns-resolution)# ns-cache-lookup enable
ACOS(config-dns-recursive-dns-resolution)# exit
ACOS(config-dns)# exit
```

2. Bind the DNS template with the SLB virtual port.

```
ACOS(config)# slb virtual-server vip-dns 192.168.10.10
ACOS(config-slb vserver)# port 53 dns-udp
ACOS(config-slb vserver-vport)# service-group group53
ACOS(config-slb vserver-vport)# template dns dns1
ACOS(config-slb vserver-vport)# exit
ACOS(config-slb vserver)# exit
```

Show Command

To view the recursive counters, use the following show command:

```
ACOS(config)# show slb virtual-server v1 53 dns-tcp application-statistics  
| inc Recursive  
Total DNS Query: 0  
Total Malformed Query: 0  
.....  
DNS Recursive Resolution Started: 0  
DNS Recursive Resolution Succeeded: 0  
Recursive Resolution NS Cache missing Glue: 7  
Recursive Resolution NS Cache entry hit: 415  
Recursive Resolution NS Cache entry miss: 25
```

DNSSEC Validation

The recursive resolver supports DNSSEC validation on responses before it continues to resolve a CNAME or sends the response to clients. It authenticates genuine DNS responses received or deny responses that cannot be trusted. This feature is disabled by default.

To enable DNSSEC validation, the following conditions must be met:

- The Checking Disabled ("CD") flag should be absent in client's query.
- The **dnssec-validation** command must be enabled and applied on per virtual port level as described in the following configuration:

```
ACOS(config)#slb template dns test  
ACOS(config-dns)#recursive-dns-resolution  
ACOS(config-dns-recursive-dns-resolution)#dnssec-validation enabled
```

Validation Workflow

The validation action provides the following results:

- If a response from a signed zone passes validation, the response contains the "AD" (Authenticated Data) flag. The AD flag is not set for a response from an unsigned zone.

- If a response from a signed zone fails validation, ACOS returns the SERVFAIL response.
- If the resolver does not support the cryptographic algorithms RRSIG/DNSKEY used or digest types of DS, ACOS removes the DNSSEC records and unsets the DO flag in the responses sent to clients.

Limitations

The recursive resolver DNSSEC validation function has the following limitations:

- It supports certain cryptographic algorithms and digest types. For the algorithms that are not supported, ACOS removes the DNSSEC records and unsets the 'DO' flag in the client response.
- It supports data size of upto 8000 bytes for DNS Record Set (RRset) validation.
- It uses DS records of the root zone as trust anchor. It does not support the configuration of the trust anchor.

For more information, see *Command Line Interface Reference*.

Supported Algorithms or Digest Types

ACOS supports the following RRSIG/DNSKEY algorithms and DNSKEY/DS digest types.

Public key's cryptographic algorithms:

Value	Description
3	DSA/SHA1
5	RSA/SHA-1
6	DSA-NSEC3-SHA1
7	RSASHA1-NSEC3-SHA1
8	RSA/SHA-256
10	RSA/SHA-512
13	ECDSA Curve P-256 with SHA-256
14	ECDSA Curve P-384 with SHA-384

For details, see www.iana.org

DS digest types:

Value	Description
1	SHA-1
2	SHA-256
4	SHA-384

For details, see www.iana.org

Recursive DNS Gateway Health Monitoring

ACOS supports health monitoring for recursive DNS gateways. This feature is useful when the recursive DNS resolver cannot serve queries due to system misconfiguration or an Internet outage. The recursive gateway's health status will also be considered when determining the state of the virtual port. This aids in the efficient management of load balancing resources.

By enabling the health monitoring, ACOS periodically performs recursive resolution based on system and template-wide configuration.

- When the recursive DNS resolution is complete and the final answer is found, the gateway is considered UP.
- When the recursive DNS resolver does not receive the final answer (or no matched type record), or an error occurs during the process, the gateway is considered DOWN.

To configure this feature, enable `recursive-dns-resolution` first and then `gateway-health-check` under SLB template DNS. In addition, the virtual port's state will also consider the result of health check if the DNS template is enabled with this feature and bound to the virtual port (`dns-udp` and `dns-tcp`).

The following health monitoring options are available:

- *interval* - Number of seconds between health check attempts.
- *retry* - Maximum number of times the DNS query retries at each server level before determining the health check failed.
- *retry-multi* - Number of times the health check consecutively fails before determining the gateway is DOWN.

- *timeout* - Period (in seconds) that recursive resolver waits for a reply to a health check.
- *up-retry* - Number of times the health check consecutively passes before determining the gateway is UP.
- *ns-cache-lookup* - Lookup NS record cache during the health check process.
- *query-name* - Query name of the probe DNS queries..
- *query-type* - Query type of the probe DNS queries..

NOTE:

The following recursive DNS resolution level configurations will be applied on the health check traffic.

- *ipv4-nat-pool* and *ipv6-nat-pool*
- *lookup-order*
- *max-trials*
- *ns-cache-lookup* (to generate NS and A/AAAA type cache from recursive resolution)

For more information, see [DNS Recursive Resolver](#).

CLI Configuration

```
ACOS(config)# slb template dns recursive_dns
ACOS(config-dns)# recursive-dns-resolution
ACOS(config-dns-recursive-dns-resolution)#gateway-health-check
ACOS(config-dns-recursive-dns-resolution-...)# interval 4
ACOS(config-dns-recursive-dns-resolution-...)# up-retry 2
```

Show Command

To view the recursive health status of the specific DNS template, use the following command:

```
ACOS(config)#show slb template dns recursive_dns recursive-dns-resolution
slb template dns name: recursive_dns
Recursive gateway status: UP
Up-retry count:          1
Down-retry count:        0
```

NOTE: To view the recursive counters of gateway health check traffic is not supported.

DNS Over HTTP/HTTPS (DoH)

Support for Incoming DNS over HTTPS (DoH) requests

ACOS now provides support for handling incoming DNS over HTTPS (DoH) requests from the client. DoH GET and POST requests conforming to RFC 8484 are supported. When a DNS over HTTPS request is received, the DNS query is extracted from the incoming HTTP request and forwarded to a backend DNS server. The HTTP traffic received in the following conditions will be evaluated for DoH processing:

- Traffic received on HTTP/HTTPS type virtual port.
- DOH Service enabled by binding doh template to virtual port.
- DNS Servers added to the DoH template for DNS query load balancing.

DNS Firewall Support with DoH

When the DoH (DNS over HTTP/HTTPS) feature is configured on ACOS, the DNS traffic decapsulated from DoH is forwarded to a DNS service-group. With the support of the service-chaining feature, the DNS traffic can be forwarded to a DNS vport on the same partition for additional DNS features such as DNS firewall, GSLB and other DNS features. However, this requires the packet to be processed twice, once for DoH and once for the DNS vport.

To avoid the dual processing, DNS firewall features can be deployed on the DoH traffic without the necessity of service-chaining.

Note:

The following set of DNS features under the DNS template are supported when using it with the DoH template:

- DNS query/response existing scrubbing features
- DNS query type/class filtering

- DNS response rate limiting
- DNS logging template
- DNS caching
- EDNS client subnet

Limitations

The following features under DNS template are not supported:

- Class-list (Class-list based rate limiting is supported at the HTTP vport level and not at the DNS packet level)
- GSLB (DNS vport feature)
- Query-id-switch
- Recursive DNS resolver
- DNSSEC service group
- Malformed-query forwarded to another service-group
- udp-retransmit (This command does not apply to DoH and is used where dns-tcp vport is bound to udp service-group. DoH UDP retransmits are config controlled in DoH template).

Known Issues and Limitations

The following templates are not supported on the virtual-port when a DoH template is bound:

- `external-service`
- `reqmod-icap`
- `respmo-icap`
- When a DoH template is bound to the virtual-port, the virtual port health status will be marked functional UP, even if there is no HTTP service-group bound to that virtual-port. This is true even if the HTTP service-group bound to the virtual-port is down.

- DoH template is supported on HTTP/HTTPS virtual-ports. It is not supported on fast-HTTP type virtual-ports.
- DoH is supported with HTTP1 as well as HTTP2.

CLI Configuration

The SLB DoH template feature is available to configure this functionality.

The “`slb template doh <name>`” command defines the DoH template. The ACOS configuration level is changed to “`config-doh`” mode.

Configure DoH as follows:

1. To configure DoH on SLB template and related parameters,

```
slb template doh doh-1
non-dns-request reject 501
conn-reuse enable
template tcp-proxy TCP1
```

2. To configure UDP and TCP service groups, use:

```
forwarder
    udp-service-group sg-53-udp
    tcp-service-group sg-53
```

NOTE: Both UDP and TCP service groups can be configured at the same time.

3. To configure IPv4 or IPv6 to forward the traffic externally or internally to another VIP. You can also set `internal` for forward the traffic internally to another VIP and/or `port` to set the forwarding port. By default, the port is 53.

```
forwarder
forwarding-ipv4 10.10.1.10 internal port 53
```

OR

```
forwarder
forwarding-ipv6 2309:db8::1 internal port 53
```

4. To configure total UDP DNS retries (max-trials), retry-interval and action to take upon UDP DNS timeout, use:

```
dns-retry
  retry-interval 100
  after-timeout retry-with-tcp
  max-trials 5
!
```

5. To bind the DoH template and to a SLB virtual-server port and enable related services like SSL on the service group.

```
slb virtual-server SVS1 1.1.1.1
  Port 80 http
    template doh doh-1
    service-group sg1
  port 443 https
    template client-ssl ssl1
    template doh doh-1
    service-group sg1
!
```

6. To view the DOH specific configuration use the following commands:

```
show slb template doh
show run slb template doh
show run / show run slb virtual-server <vip-name>
```

7. To view the DOH counters use the following commands:

```
show slb http-proxy detail | i DOH
show slb http-proxy debug | i DOH
show slb http-proxy <virtual-server-name> <port_number> | i DOH
show slb http-proxy <virtual-server-name> <port_number> detail | i DOH
```

GUI Configuration

The following section provides details on how to configure DoH template on SLB, associate it to an SLB virtual port, and perform the related configurations through ACOS GUI. For details on the field names and parameters, refer to *Online Help* on ACOS GUI.

Add DoH Template

1. Navigate to **ADC>> Template >> L7 Protocols** page to configure the DoH template options.
2. Click **+Create** and in the drop-down menu, select **DNS over HTTPs** template.

The **Create DNS Over HTTP Template** window opens up. Enter a name for the template, and all the configuration parameters. Click **Save** to create the new DoH template.

Associate DoH template to SLB Virtual Port

1. Navigate to **ADC > Template > Create > Virtual Port** page.
2. On the **Virtual Port** page, a new option, **Template DOH** is displayed with a drop-down selection, listing the pre-defined DOH templates for HTTP and HTTPS protocols.

DoH Statistics on HTTP Proxy Page.

Once DoH is configured, to view the DoH related HTTP statistics:

1. Navigate to **ADC >> Statistics >> HTTP Proxy** page.
2. Select **Virtual Server** and **HTTP Ports options** and view the related DoH counters and statistics.

DNS Filtering Using Web Categories

This section provides an overview of DNS Filtering using Web Categories, the configurable parameters, and how to configure it.

The following topics are covered:

Overview	674
DNS Filtering Actions	675
Additional Options	676
DNS Filtering Outcomes	677
CLI Configuration	678

Overview

DNS filtering is a technique that prevents users from accessing specific websites or content on the internet. Using this technique, requests to specific domains or websites can be blocked based on predefined criteria such as social media, adult content, or gaming sites.

DNS Filtering can be implemented effectively in the following scenarios:

- To protect against potential security risks by blocking access to malicious websites that host malware, phishing scams, or other cyber threats.
- To comply with certain regulations or corporate policies regarding internet usage. For example, blocking access to social media sites during work hours.
- To enforce parental control by restricting children's access to certain websites or content on the internet.

ACOS has leveraged Webroot's website classification data to implement a flexible and scalable DNS filtering technique. Webroot provides a comprehensive set of web categories that cover a wide range of content types, such as social networking, gaming, adult content, malware, travel, banking, and more.

The web categories allow configuring DNS filtering policies based on the content type rather than a list of individual websites. ACOS DNS filtering checks the requested domain against Webroot's web categories and then decides the action to be taken.

NOTE:	You need to download and install the Webroot's web classification database on the system. The Webroot license must be installed and enabled before utilizing the database. For detailed steps, refer to the <i>SSL Insight (SSLi) Configuration</i> guide and search for 'Web Category License'.

Workflow

1. The DNS requests are sent to the DNS filter.
2. The DNS filter checks the requested domain against Webroot's web categories.
3. If the domain falls in the allowed category, the DNS request is permitted to proceed for DNS resolution.

4. If the domain falls in the denied (blocked) category, the DNS request is either dropped or redirected, based on the action configured in the policy.

DNS Filtering Actions

The `category-lookup` command under the SLB DNS template allows you to configure actions after the DNS request matches the web categories. Consider the following configuration example:

```
slb template dns dns_temp
  category-lookup web1 permit
  category-lookup web2 drop
!
```

In this case, all the DNS requests that match the web categories in `web1` list are permitted, whereas the requests that match the web categories in `web2` list are dropped.

Using the `category-lookup` command, you can configure one of the following actions:

- **permit** – The DNS request is permitted to proceed with DNS resolution. This is the default action.
- **drop** – The DNS request is dropped.
- **respond** – ACOS replaces the response with the configured parameters. The following parameters are available:
 - **IP** – ACOS responds with the specified IPv4 address in case of A record request. However, it does not respond to other records such as AAAA, CNAME, MX, NS, TXT, or PTR.
 - **IPv6** – ACOS responds with the specified IPv6 address in case of AAAA record query. However, it does not respond to other records such as A, CNAME, MX, NS, TXT, or PTR).
 - **CNAME** – ACOS responds with the specified CNAME for all record type requests. The CNAME record is used to create an alias for a domain.
 - **IP**, **IPv6**, and **CNAME** (all three configured together) – ACOS responds with an IPv4

address for A record request, an IPv6 address for AAAA request, and the configured CNAME for other records such as CNAME, MX, NS, TXT, or PTR.

- **NXDOMAIN** – ACOS responds with the NXDOMAIN record to all record types. NXDOMAIN record indicates that the domain name is not registered or does not exist.

NOTE:

- Configuring the same category list with different actions is prohibited.
 - Configuring the response **NXDOMAIN** with other actions is prohibited.
-

Additional Options

- **category-lookup-bypass** *dns-type-list* - Specify a DNS type class-list containing a list of domains to be bypassed during DNS filtering. Use the **class-list** command to configure the DNS type class list. The domains that match the class-list will not be filtered out during web category lookup and will continue with DNS resolution.

NOTE:

Only class-lists of type DNS can be specified.

- **category-lookup-online-lookup** - Configure this option to enable online lookup i.e., connect to the Webroot's server for domain matching.

By default, online lookup is disabled during web category lookup, i.e., the system first checks the locally downloaded web classification database for domain matches. If the requested domain name is not found in the database, the query is uncategorized and proceeds with DNS resolution (based on the policy). An online lookup is initiated after this request is processed to ensure that the local database is updated and includes the previously uncategorized domain. Therefore, subsequent requests to this uncategorized domain would be categorized correctly.

When the **category-lookup-online-lookup** option is enabled, if the request is not found in the database, online lookup is initiated immediately. The DNS request is categorized, and the action is taken based on DNS template configuration.

DNS Filtering Outcomes

- If a DNS request matches more than one category, the action is based on the priority, with the **permit** action taking precedence over **drop** and **respond**.
- If a DNS request matches two categories that have the **respond** action configured, the web list with action **CNAME** record takes precedence.
- If a DNS request matches two categories that have the **CNAME** action configured, the configuration order determines the action.
- If **CNAME** action is not configured for unavailable DNS record types, clients querying for those record types receive an empty response (no answer).
- If a DNS request matches a class-list to be bypassed and also matches a category that has **drop/respond** action configured, the request will be bypassed and will proceed for DNS resolution.
- If a DNS request matches a class-list to be bypassed when an aFlex is configured, the aFlex script is executed since it takes precedence.
- If a DNS request does not match any category, it is placed in the **uncategorized** web category and then sent for DNS resolution.

Example Configurations

Example 1

```
slb template dns dns
category-lookup web1 response IPV6 3:3:3::3
category-lookup web2 response IP 2.2.2.2 cname www.abc.com
```

Outcomes with different DNS requests:

- In case of AAA record request that matches categories in **web1**, ACOS responds with the IPv6 address 3:3:3::3.
- In case of an A record request that matches categories in **web2**, ACOS responds with the IPv4 address 2.2.2.2.
- In case of an MX record request that matches categories in **web2**, ACOS responds with CNAME **www.abc.com**.

Example 2

```
slb template dns dns
category-lookup web1 response IP 3.3.3.3
category-lookup web2 response IP 2.2.2.2 cname www.abc.com
```

In this case, if the DNS request matches categories in both web lists, the web list with the action **cname** in the response takes precedence. Therefore, for an A record request, ACOS responds with the IPv4 address 2.2.2.2, and for an MX record request, ACOS responds with the CNAME www.abc.com.

Example 3

```
slb template dns dns
category-lookup web1 response IP 3.3.3.3 cname www.xyz.com
category-lookup web2 response IP 2.2.2.2 cname www.abc.com
```

In this case, if the DNS request matches categories in both web lists, the configuration order determines the action, since both the lists have **cname** action configured. Therefore, for an A record request, ACOS responds with the IPv4 address 3.3.3.3, and for an MX record request, ACOS responds with the CNAME www.xyz.com.

CLI Configuration

NOTE: DNS filtering using Webroot categories can also be configured using aFlex. However, the DNS template configuration takes precedence over aFlex, meaning ACOS checks and executes the DNS template configurations before aFlex. Consequently, if the template configuration executes the **respond** or **drop** action, the request is not processed further i.e., aFlex processing is bypassed. However, if the template configuration permits the request, ACOS proceeds with the aFlex configuration.

- To configure DNS filtering, you need to create a web category list and then configure the actions to be taken when the DNS request matches a category from the list.

Perform the following steps to configure DNS filtering using web categories:

1. Configure the global DNS settings.

```
ACOS(config)# slb common
ACOS(config-common)# dns-negative-cache enable
ACOS(config-common)# dns-cache-enable
ACOS(config-common)# dns-cache-age 3000
ACOS(config-common)# dns-cache-entry-size 4096
ACOS(config-common)# honor-server-response-ttl
```

2. Create the web category lists using the **web-category** command.

```
ACOS(config)# web-category
ACOS(config-web-category)# use-mgmt-port
ACOS(config-web-category)# enable
ACOS(config-web-category)# category-list web1
ACOS(config-web-category-list)# computer-and-internet-info
ACOS(config-web-category-list)# search-engines
ACOS(config-web-category-list)# exit
ACOS(config-web-category)# category-list web2
ACOS(config-web-category-list)# business-and-economy
```

NOTE: For the complete list of web categories supported, refer to the **web-category** command description in the *Command Line Interface Reference* guide.

3. Configure a recursive DNS resolver.

```
ACOS(config)# slb template dns dns_recursive
ACOS(config-dns)# recursive-dns-resolution
ACOS(config-dns-recursive-dns-resolution)# ipv4-nat-pool snat
ACOS(config-dns-recursive-dns-resolution)# full-response
ACOS(config-dns-recursive-dns-resolution)# exit
```

4. Configure the **category-lookup** command under the SLB DNS template to set the actions to be taken when a DNS request matches a category from the web lists. For more information on actions, see [DNS Filtering Actions](#). Additionally, to enable online lookup, configure the **category-lookup-online-lookup** command.

```
ACOS(config-dns)# category-lookup-online-lookup
ACOS(config-dns)# category-lookup web1 permit
```

```
ACOS(config-dns)# category-lookup web2 respond IP 2.2.2.200
```

NOTE:

- The **category-lookup** command applies to all types of DNS (including Recursive Resolver).
- Up to eight category lists can be configured under the same DNS template

5. Bind the DNS template *dns_recursive* (configured in the previous step) with the SLB virtual port.

```
ACOS(config)# slb virtual-server vs1 192.168.100.20
ACOS(config-slb vs1)# port 53 dns-tcp
ACOS(config-slb vs1-vport)# source-nat auto
ACOS(config-slb vs1-vport)# template dns dns_recursive
ACOS(config-slb vs1-vport)# exit
ACOS(config-slb vs1)# port 53 dns-udp
ACOS(config-slb vs1-vport)# source-nat auto
ACOS(config-slb vs1-vport)# template dns dns_recursive
```

- To bypass a specific domain during web category lookup, create a DNS type class list before configuring the SLB DNS template.

Perform the following steps:

1. Create a DNS type class-list by specifying a match-criteria for the domains to be bypassed.

```
ACOS(config)# class-list my-list dns
ACOS(config-class list)# dns contains apple
```

In this case, any domain that contains the term 'apple' will be added to the class-list.

2. While configuring the SLB DNS template, specify the DNS type class list using the **category-lookup-bypass** command.

```
ACOS(config)# slb template dns dns_temp
ACOS(config-dns)# category-lookup-bypass my-list
ACOS(config-dns)# category-lookup web1 permit
```

```
ACOS(config-dns)# category-lookup web2 drop
```

As per the configuration, any domain name that contains the term 'apple' will be bypassed and will continue with DNS resolution.

Show Commands

- To view the counters specific to a web category in the category list, execute the following show command:

```
ACOS(config)# show counters web-category category-list web1
show counters web-category category-list web1
-----
uncategorized category                                0
real estate category                                  0
computer and internet security category                0
financial services category                            0
business and economy category                          0
computer and internet info category                    2
```

- To view DNS queries' statistics based on web categories, execute the following show command:

```
ACOS(config)# show slb virtual-server vip 53 dns-udp application-
statistics | inc Category
Category Action Drop:                                0
Category Action Respond:                              2
Category Action Permit:                              0
Category Respond NXDomain:                            0
Category Respond No Answer:                          0
Category Respond Type A:                              1
Category Respond Type AAAA:                          0
Category Respond Type CNAME:                         1
Category Bypass:                                      0
Category Async Request Sent (Template + aFlex):      1
Category Async Result (Template + aFlex):            1
Category No Local No Lookup (Template + aFlex):      0
```

- To check if a domain belongs to a web category present in the local database, execute the following command:

```
ACOS(config)# show web-category url-category <url-name> local-db-only
```

RAM Caching

This section describes usage of an ACOS device as a RAM cache server.

The following topics are covered:

Overview	683
Configuring RAM Caching	688

Overview

The RAM Cache is a high-performance, in-memory Web cache that by default caches HTTP responses (RFC 2616 compliant). The RAM Cache can store a variety of static and dynamic content and serve this content instantly and efficiently to a large number of users.

Caching HTTP content reduces Web server transactions and hence the server loads. Caching dynamic content reduces latency and computation cost of generating dynamic pages by application servers and database servers. Caching also reduces page download time and bandwidth utilization.

RAM caching is useful for high-demand objects on a website, for static content such as images, and when used in conjunction with compression to store compressed responses, eliminating unnecessary overhead.

RFC 2616 Support

In general, ACOS RAM caching conforms with the cache requirements described in RFC 2616,

“Hypertext Transfer Protocol -- HTTP/1.1”, in sections 13 and 14.

ACOS RAM caching considers HTTP responses with the following status codes to be cacheable. However, if there is no Content-Length header, the response will not be cached. Additionally, warning headers are not supported.

- 200 – OK
- 203 – Non-Authoritative Response
- 300 – Multiple Choices
- 301 – Moved Permanently
- 302 – Found (only if Expires header is also present)
- 410 – Gone

If-Modified-Since Header Support

ACOS supports If-Modified-Since (IMS) headers in GET requests from clients. Optionally, a client can include the following header in a GET request:

```
If-Modified-Since: date-time
```

This header instructs the content server (or cache server) to send the requested page only if the page has been modified subsequent to the date and time specified in the IMS header.

ACOS responds as follows:

- If the requested content is in the cache and is still fresh, but the content was cached before the date and time in the IMS header, the ACOS device sends a 304 – Not Modified response to the client.
- If the requested content is in the cache and is still fresh, and the content was cached after the date and time in the IMS header, the ACOS device sends a 200 – OK response, along with the requested page, to the client.
- If the requested content is not in the cache, or is in the cache but is stale, the ACOS device deletes the IMS header from the request. This forces the server to send a 200 OK response, which is then immediately cached.

Support for no-cache and max-age=0 Cache-Control Headers

According to RFC 2616, either of the following Cache-Control headers in a request should make the cache (the ACOS device) reload the cached object from the origin server:

- Cache-Control: no-cache
- Cache-Control: max-age=0

However, for security, support for these headers is disabled by default. These headers can make the ACOS device vulnerable to Denial of Service (DoS) attacks.

To enforce strict RFC compliance, you can enable support for the headers.

Age and Via Headers into Cached Responses

RAM caching supports Age and Via header insertion into cached server replies before they are sent to clients.

- Age header – indicates the age of the cached response, measured in seconds
- Via header – indicates the ACOS software version, in the following format:

```
ACOS-CACHE-software-version(major.minor):last-octet-of-VIP address
```

Here is an excerpt of a cached server reply containing these headers:

```
HTTP/1.1 200 OK
Server: ACOS-3200
Date: Thu, 04 Mar 2010 20:46:23 GMT
Content-Type: text/plain
Content-Length: 4096
Last-Modified: Fri, 29 Jan 2010 00:37:46 GMT
Age: 230
Via: ACOS-CACHE-2.4:130
```

Insertion of these headers is enabled by default. You can disable insertion of the Age and Via headers, in individual RAM caching templates.

ACOS RAM Cache Behavior

This section summarizes the behavior of ACOS RAM caching when this feature is enabled.

When Processing the Request:

- If a cache policy is configured, ACOS checks if the URI in the request matches any of the URIs configured for the cache policy. If there is a match, the configured action (**invalidate**, **cache**, **nocache**) is remembered for that request.
- If there is no URI match, ACOS checks to see if **default-policy-nocache** is configured; if it is configured, the request is marked as not cacheable.

When Processing the Response to a Request Received from the Server:

- ACOS checks to see if response should be cached based on request processing results.
- If the response is cacheable, ACOS ignore the Cache-Control from server response.
- In summary, the server's cache-control header in the HTTP response can override the cache policy. The `default-policy-nocache` only applies when there is a cache policy configured but there is no URI match.

Some additional implementation details are provided below:

- A response for a request that uses any method other than GET is not cached.
- A response for a GET request that contains a body is not cached.
- A request that contains an Authorization or a Proxy-Authorization header is not cacheable. The authorization header contains security-related information and should not be cached.
- A response for a request that contains an If-Match header or an If-Unmodified-Since header is not cached.
- An HTTP response which has a Vary header with a value of anything other than Accept-Encoding is not cached. (The compression module inserts the Vary: Accept-Encoding header.)
- A response with a Cache-Control header with one of the following values: No-Cache, No-Store, Private is not cached.
- If the Cache-Control header has one of the following values: Public, Must-Revalidate, Proxy-Revalidate, max-Age, S-maxage it is cached.
- Responses that contain a Pragma header are not cached.
- Responses that contain a Set-Cookie or a Set-Cookie2 header are not cached. (Responses with Cookie headers often contain information that is specific to the user and so should not be cached.)
- If the response type is FIN terminated, or the response does not have one of the following attributes: Content-Length, or Transfer-Encoding: Chunked, it is not cached.
- ACOS RAM caching does not cache responses that contain cookies. In configurations that also use cookie persistence, this behavior prevents server responses from being cached. You can enable the ACOS device to remove cookies

from server replies, so that the replies can be cached.

- Image files are an exception; RAM caching can cache images that have cookies.

Dynamic Caching

You can enhance RAM caching performance with dynamic RAM caching. Dynamic RAM caching is useful in situations where the response to a client request can be used multiple times before the response expires. Here are some examples where dynamic RAM caching is beneficial:

- The same response is usable by multiple users within a certain period of time. In this case, dynamic RAM caching is useful even if the cache expiration period is very small, if enough users access the response within that period. For example, dynamic RAM caching is beneficial for a hierarchical directory that is generated dynamically but presents the same view to all users that request it.
- The response is usable by only a single user but the user accesses it multiple times. For example, if the response generated in one session can be used unchanged in a second session.

Host Verification

RAM caching has an optional host verification feature. Host verification supports multiple name-based virtual hosts. Name-based virtual hosts are host names that share the same IP address. For example, the real server IP address 192.168.209.34 could be shared by the following virtual hosts:

- www.abc.com
- www.xyz.com

By default, host verification is disabled. When the ACOS device receives the server response for cacheable content, the ACOS device caches the content along with the URI, but not the host name. For example, if a client requests `http://www.abc.com/index.html`, the ACOS device caches the content and `/index.html` but does not cache `abc.com`. If another request is received, for `http://www.xyz.com/index.html`, the ACOS device serves the same content.

If you enable host verification, the ACOS device caches the host name along with the URI. For example, for `http://www.abc.com/index.html`, the ACOS device caches the content, `/index.html`, and `abc.com`. If a new request is received, for `http://www.xyz.com/index.html`, the ACOS device checks the cache for content indexed by both `/index.html` and `xyz.com`. ACOS serves the content to the client only if the content was cached for `xyz.com`.

Configuring RAM Caching

To configure RAM caching:

1. Add the real servers that serve the content to be cached, if not already configured.
2. Configure a service group and add the real servers to it, if not already configured.
3. Configure a cache template with settings for the type and size of content to be cached. Optionally, configure dynamic caching policies.
4. Configure the virtual server, and bind the service group and cache template to the service ports for which caching will be provided.

NOTE: RAM cache is not supported when the `respmo-icap` template is utilized.

The following examples are provided:

Basic RAM Caching Configuration (CLI Example)

The commands in this example enable RAM caching for virtual service port TCP 80 on VIP `cached-vip`.

These commands add a RAM caching template using default template settings.

```
ACOS(config)# slb template cache ramcache
ACOS(config-ram caching)# disable-insert-age
ACOS(config-ram caching)# disable-insert-via
ACOS(config-ram caching)# exit
```

The following commands configure the real servers.

```
ACOS(config)# slb server RS1 192.168.90.34
```

```
ACOS(config-real server)# port 80 tcp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# port 443 tcp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
ACOS(config)# slb server RS2 192.168.90.35
ACOS(config-real server)# port 80 tcp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# port 443 tcp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
```

The following commands configure the service group.

```
ACOS(config)# slb service-group cached-group tcp
ACOS(config-slb svc group)# member RS1 80
ACOS(config-slb svc group-member:80)# exit
ACOS(config-slb svc group)# member RS1 443
ACOS(config-slb svc group-member:443)# exit
ACOS(config-slb svc group)# member RS2 80
ACOS(config-slb svc group-member:80)# exit
ACOS(config-slb svc group)# member RS2 443
ACOS(config-slb svc group-member:443)# exit
ACOS(config-slb svc group)# exit
```

The following commands configure the virtual server and bind the RAM caching template and the service group to virtual HTTP service port 80.

```
ACOS(config)# slb virtual-server cached-vip 10.10.10.101
ACOS(config-slb vserver)# port 80 http
ACOS(config-slb vserver-vport)# service-group cached-group
ACOS(config-slb vserver-vport)# template cache ramcache
ACOS(config-slb vserver-vport)# exit
ACOS(config-slb vserver)# exit
```

This command shows client sessions. Asterisks (*) in the Reverse Source and Reverse Dest fields indicate the ACOS device directly served the requested content to the client from the ACOS RAM cache. In this case, the session is actually between the client and the ACOS device rather than the real server.

```
ACOS(config)# show session
```

Traffic Type		Total	

TCP Established		4328	
TCP Half Open		39026	
UDP		0	
Non TCP/UDP IP sessions		0	
Other		0	
Reverse NAT TCP		0	
Reverse NAT UDP		0	
Free Buff Count		0	
Curr Free Conn		1923655	
Conn Count		5287134	
Conn Freed		5113720	
tcp syn half open		0	
Prot	Forward Source	Forward Dest	Reverse Source
Reverse Dest Age			

Tcp	10.10.10.61:25058	10.10.10.10:80	* *
	600		
Tcp	10.10.10.60:9239	10.10.10.11:80	* *
	600		
Tcp	10.10.10.61:1838	10.10.10.10:80	* *
	600		
Tcp	10.10.10.65:47834	10.10.10.11:80	* *
	600		
Tcp	10.10.10.62:55613	10.10.10.11:80	* *
	600		
Tcp	10.10.10.57:9233	10.10.10.11:80	* *
	600		

The following command shows RAM caching statistics.

```
ACOS(config)# show slb cache
```

```
Total
```

```
-----
Cache Hits          0
```

```

Cache Misses          6
Memory Used           27648
Bytes Served           0
Entries Cached         6
Entries Replaced       0
Entries Aged Out       0
Entries Cleaned        0
Total Requests         0
Cacheable Requests     0
No-cache Requests      0
No-cache Responses     0
IMS Requests           0
304 Responses          0
Revalidation Successes 0
Revalidation Failures  0
Policy URI nocache     0
Policy URI cache       0
Policy URI invalidate  0
Content Too Big        0
Content Too Small      0
Srvr Resp - Cont Len   220
Srvr Resp - Chnk Enc   37
Srvr Resp - 304 Status 0
Srvr Resp - Other      0
Cache Resp - No Comp   383579
Cache Resp - Gzip       0
Cache Resp - Deflate    0
Cache Resp - Other      0
Entry create failures  0

```

The following command shows cached objects.

```
ACOS# show slb cache entries cached-vip 80
```

```
cached-vip:80
```

Host	Object URL	Bytes	Type
Status	Expires in		

10.20.0.130	/16k.html	16744	CL, No FR
165 s			

10.20.0.130	/4k.html	4303	CL, No	FR
166 s				
10.20.0.130	/32k.html	32976	CE, No	FR
169 s				
10.20.0.130	/1024k.html	108786	CL, Gz	FR
162 s				
10.20.0.130	/8k.html	8399	CE, No	FR
165 s				
10.20.0.130	/64k.html	65744	CE, Gz	FR
168 s				

The Status column indicates the status. In this example, all entries are fresh (FR). For more information, see the *Command Line Interface Reference*.

Dynamic Caching Configuration (CLI Example)

Here is an example application of dynamic RAM caching. Web site x.y.com displays a frequently requested list page and serves private pages to individual clients based on additional requests from clients. Clients also can add or delete content on the list page.

```
http://x.y.com/list
http://x.y.com/private?user=u1
http://x.y.com/add?a=p1&b=p2
http://x.y.com/del?c=p3
```

Dynamic RAM caching policies can be used to effectively manage caching for this site.

The /list URI is visited by many users and therefore should be cached, so long as the content is current. However, the /private URI contain private data for a specific user, and should not be cached.

The /add and /del URLs modify the content of the list page. When either type of URI is observed by the ACOS device, the currently cached content for the /list URI should be invalidated, so that new requests for the URI are not served with a stale page.

The following commands implement the dynamic RAM caching configuration described above.

```
ACOS(config)# slb template cache ram-cache
```



```
ACOS(config-ram caching)# policy uri /list cache 3000
ACOS(config-ram caching)# policy uri /private nocache
ACOS(config-ram caching)# policy uri /add invalidate /list
ACOS(config-ram caching)# policy uri /del invalidate /list
ACOS(config)# exit
```

The policy that matches on “/list” caches content for 50 minutes. The policy that matches on “/private” does not cache content. Policies that match on “/add” and “/del” invalidate the cached “/list” content.

Configuration To Flush Specific Cache Entries (CLI Example)

To flush specific entries from the RAM cache, use an invalidate policy as shown by this example:

```
ACOS(config)# slb template cache ram-cache
ACOS(config-ram caching)# policy uri /story cache 3600
ACOS(config-ram caching)# policy uri /flush invalidate /story
ACOS(config-ram caching)# exit
```

This policy is configured to flush (invalidate) all cached entries that have “/story” in the URI. The policy is activated when a request is received with the URI “/flush”.

Transparent Cache Switching

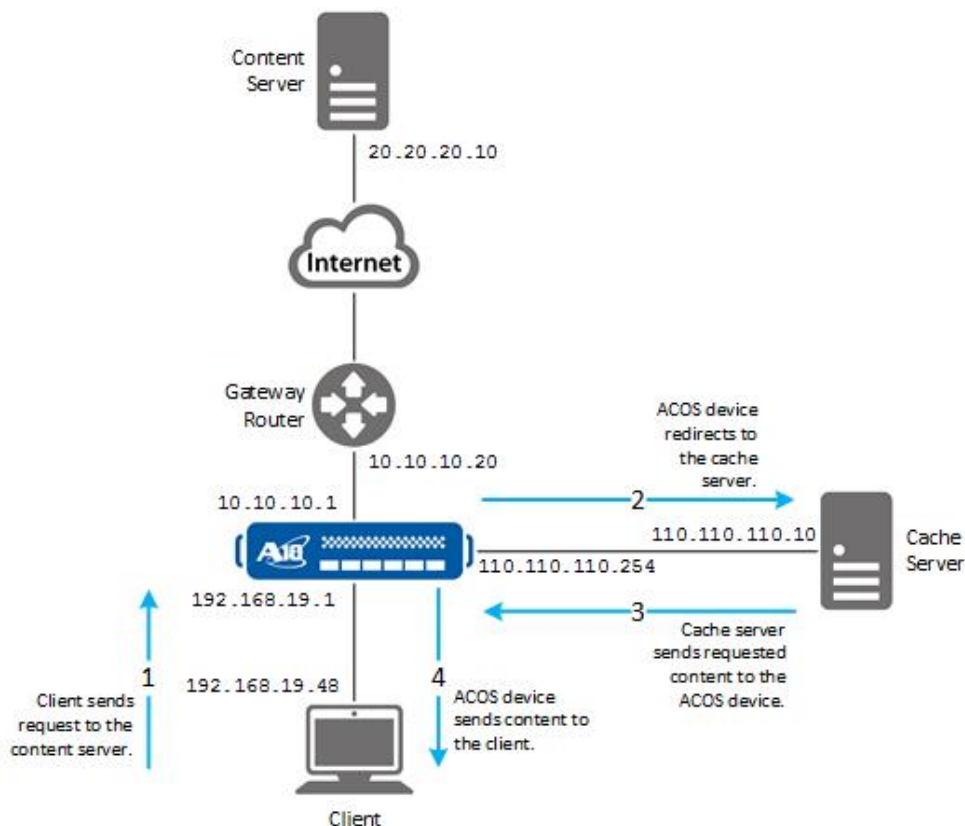
The following topics are covered:

- [Overview](#) 695
- [Configuring Transparent Cache Switching](#) 698

Overview

ACOS supports Transparent Cache Switching (TCS). TCS improves server response times by redirecting client requests for content to cache servers containing the content. [Figure 65](#) shows an example topology.

Figure 65 : Transparent Cache Switching



In this example, a client sends a request for content that is hosted by the content server. ACOS redirects the client's request to the cache server. If the cache server has the requested content, the cache server sends the content to the ACOS device, which sends the content to the client.

If the content is cacheable, but the cache server does not have the requested content or the content is stale, the cache server requests the content from the content

server, caches the content, then sends the content to the ACOS device, which sends the content to the client.

TCS Granularity

You can configure Layer 4 TCS or Layer 7 TCS.

- Layer 4 TCS – Sends all TCP or UDP traffic addressed to the content server to the cache server instead
- Layer 7 TCS – You can configure Layer 7 TCS with either of the following levels of granularity:
 - Sends all HTTP requests to the cache server and sends all other requests to the content server
 - Sends HTTP requests for specific URLs to the cache server; sends other requests to the content server

Using Multiple Cache Servers

If your network uses multiple cache servers, you can configure destination-IP persistence, to always select the same cache server for content from a given destination IP address. This technique reduces cache misses, by ensuring that requests for a given site IP address always go to the same cache server.

For even greater control, you can configure the ACOS device to select from among multiple cache service groups based on the requested URL. When combined with destination-IP persistence, this method allows you to control initial selection of the cache service group, after which the ACOS device always sends requests for the same content to the same cache server within the cache service group.

Application Templates

TCS does not require configuration of any application templates. However, you can use the following types of application templates for advanced features, such as URL-based Layer 7 TCS:

- HTTP template – To selectively redirect client requests based on URL strings, use an HTTP template containing URL switching rules. When a client request matches the URL string in a URL switching rule, the ACOS device selects the service group specified in the URL switching rule, instead of the service group bound to the virtual port.

For example, you can configure a URL switching rule that matches on any URL that contains “.mycorp/”. In this case, requests for any URL that contains “.mycorp/” are sent to the service group that contains the cache server. Requests for other URLs are sent to the gateway router instead.

In Layer 7 TCS configurations using URL switching, a separate real server is required for the gateway router, and the real server must be placed in its own service group. The gateway router’s service group is the default service group for the virtual port. Client requests to a URL that does not match a URL switching rule are sent to the gateway router’s service group instead of the cache server’s service group.

- Destination-IP persistence template – In deployments with multiple cache servers, use a destination-IP persistence templates to ensure the same cache server is used for every content request from a given content server. ACOS uses standard SLB to select a cache server for the first request to a real server IP address and assigns a hash value to the server. Subsequent requests for the same real server are sent to the same cache server.

By always using the same cache server for content from a given server, a destination-IP persistence template can reduce duplication of content on multiple cache servers, and can also reduce cache misses.

- RAM caching template – To also cache some content on the ACOS device itself, you can use a RAM caching template. In this case, the ACOS device directly serves content that is cached on the ACOS device, and only sends requests to the cache server for content that is not cached on the ACOS device.
- Connection reuse template – Use connection reuse templates to reuse TCP connections. When a client session ends, the TCP connection is not terminated; the connection is reused for a new client session.

Spoofing Caches

Some cache servers can use the client's IP address instead of the cache server's IP address as the source address when obtaining content requested by the client. A cache server operating in this mode is a spoofing cache server. Configuration for a spoofing cache server includes a couple of additional steps. (See [Enabling Support for Cache Spoofing](#).)

High Availability Support

You can deploy TCS in VRRP-A high availability configurations. For an example of TCS deployed in Layer 3 inline mode of HA, see [Configuring TCS: VRRP-A High Availability](#).

Configuring Transparent Cache Switching

This section describes the Transparent Cache Switching (TCS) configuration procedures and examples.

The following topics are covered:

Configuring Layer 4 TCS	699
Configuring Layer 7 TCS	702
Optimizing TCS with Multiple Cache Servers	707
Enabling Support for Cache Spoofing	709
Configuring TCS: VRRP-A High Availability	710
Configuring TCS for FTP	727
Configuring Bandwidth Limits Per Server or Port	730
Configuring Destination Hash-Based TCS	734

Configuring Layer 4 TCS

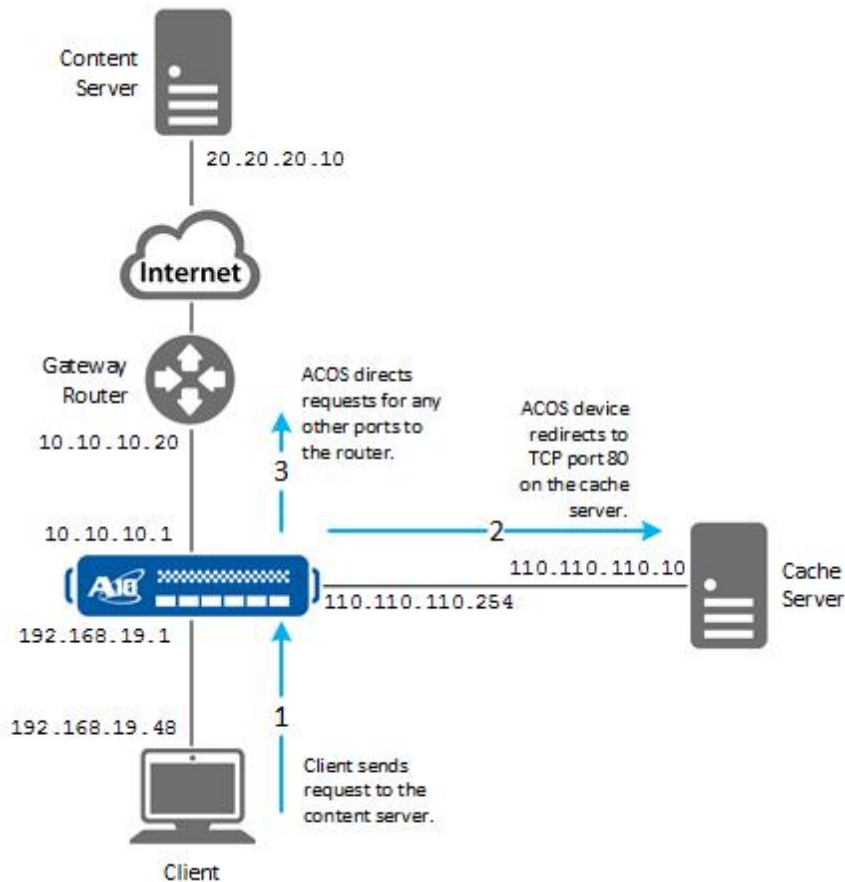
Configuring Layer 4 TCS (Procedure)

1. Configure the interfaces connected to the clients, the content servers, and the cache server. Enable promiscuous VIP on the ACOS interface(s) connected to the clients.
2. Configure an extended ACL that uses the permit action and that matches on client addresses as the source address, and on the content server address as the destination address.
3. Configure a real server for the cache server. Add the TCP or UDP port; for example, TCP port 80.
4. If the cache server spoofs client IP addresses when requesting content from content servers, enable cache spoofing support.
5. Configure a service group for the cache server and add the cache server to it.
6. Configure a virtual server with virtual IP address 0.0.0.0 (the wildcard VIP address) and bind it to the ACL.
7. Add virtual port 80 and bind it to the service group containing the cache server. Disable destination NAT on the virtual port.
8. If the cache server spoofs client IP addresses when requesting content from content servers, enable cache spoofing on the ACOS interface connected to the cache server and on the real (cache) server.

Configuring Layer 4 TCS (CLI Example)

The commands in this section implement the TCS configuration shown in [Figure 66](#).

Figure 66 : Layer 4 TCS



These commands configure the ACOS interface to the client. Promiscuous VIP is enabled on the interface.

```
ACOS(config)# vlan 4
ACOS(config-vlan:4)# tagged ethernet 4
ACOS(config-vlan:4)# router-interface ve 4
ACOS(config-vlan:4)# exit
ACOS(config)# interface ve 4
ACOS(config-if:ve:4)# ip address 192.168.19.1 255.255.255.0
ACOS(config-if:ve:4)# ip allow-promiscuous-vip
ACOS(config-if:ve:4)# exit
```

The following commands configure the ACOS interface to the content server.

```
ACOS(config)# vlan 2
ACOS(config-vlan:2)# tagged ethernet 2
ACOS(config-vlan:2)# router-interface ve 2
```



```
ACOS(config-vlan:2) # exit  
ACOS(config) # interface ve 2  
ACOS(config-if:ve:2) # ip address 10.10.10.1 255.255.0.0  
ACOS(config-if:ve:2) # exit
```

The following commands configure the interface to the cache server:

```
ACOS(config) # interface ethernet 5  
ACOS(config-if:ethernet:5) # ip address 110.110.110.254 255.255.255.0  
ACOS(config-if:ethernet:5) # exit
```

This command configures an extended ACL to match clients and the content server. The ACL in this example matches on any source address (client IP address) and on the destination IP address of the content server.

```
ACOS(config) # access-list 198 permit ip any host 20.20.20.10 log
```

The following commands configure a real server for the cache server. TCP port 80 is added to the real server.

```
ACOS(config) # slb server cache-rs 110.110.110.10  
ACOS(config-real server) # port 80 tcp  
ACOS(config-real server-node port) # exit  
ACOS(config-real server) # exit
```

The following commands configure a service group for the cache server:

```
ACOS(config) # slb service-group sg-tcs tcp  
ACOS(config-slb svc group) # member cache-rs 80  
ACOS(config-slb svc group-member:80) # exit  
ACOS(config-slb svc group) # exit
```

The following commands configure a wildcard VIP and bind it to the ACL:

```
ACOS(config) # slb virtual-server wildcard 0.0.0.0 acl 198  
ACOS(config-slb vserver) # port 80 tcp  
ACOS(config-slb vserver-vport) # service-group sg-tcs  
ACOS(config-slb vserver-vport) # no-dest-nat  
ACOS(config-slb vserver-vport) # exit  
ACOS(config-slb vserver) # exit
```

Configuring Layer 7 TCS

Layer 7 TCS can be configured in either of the following ways. Select one of these methods based on the level of granularity you want to use for traffic redirection.

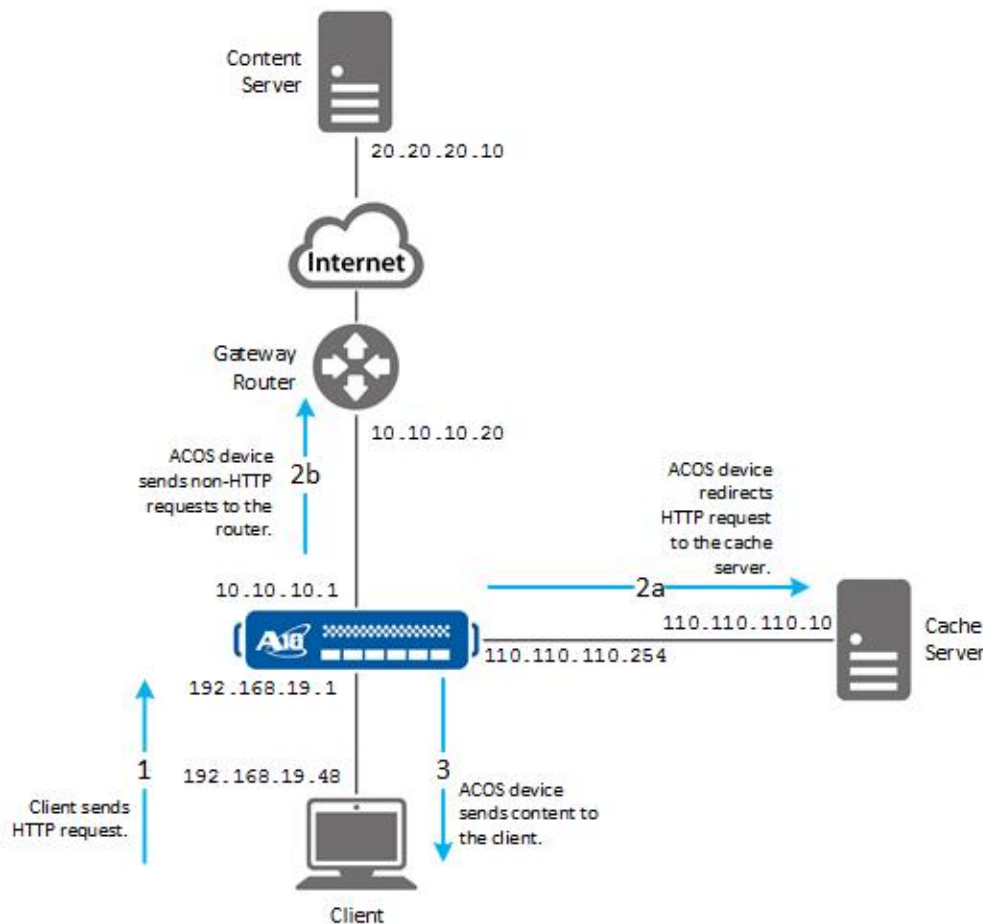
The following topics are covered:

Service Type HTTP Without URL Switching Rules	702
Service Type HTTP with URL Switching Rules	704

Service Type HTTP Without URL Switching Rules

The Service type HTTP without URL switching rules method redirects all HTTP traffic to the cache server. The configuration steps are very similar to those for Layer 4 TCS. The only difference is use of HTTP instead of TCP or UDP as the service type of the virtual port. [Figure 67](#) shows an example of this method.

Figure 67 : Layer 7 TCS Without URL Switching Rules



Service Type HTTP Without URL Switching Rules (Procedure)

1. Configure the interfaces connected to the clients, the content servers, and the cache server. Enable promiscuous VIP on the ACOS interface(s) connected to the clients.
2. Configure an extended ACL that uses the permit action and that matches on client addresses as the source address, and on the content server address as the destination address.
3. Configure a real server for the cache server. Add the TCP port; for example, TCP port 80.
4. Configure a service group for the cache server and add the cache server to it.

5. Configure a virtual server with virtual IP address 0.0.0.0 (the wildcard VIP address) and bind it to the ACL.
6. Add virtual port 80 with service type HTTP and bind it to the service group containing the cache server. Enable/disable destination NAT on the virtual port.

Service Type HTTP Without URL Switching Rules (CLI Example)

The commands in this section implement the TCS configuration shown in [Figure 67](#). The commands are as those used in the Layer 4 TCS example, and are therefore not shown.

The following commands configure a wildcard VIP and bind it to the ACL:

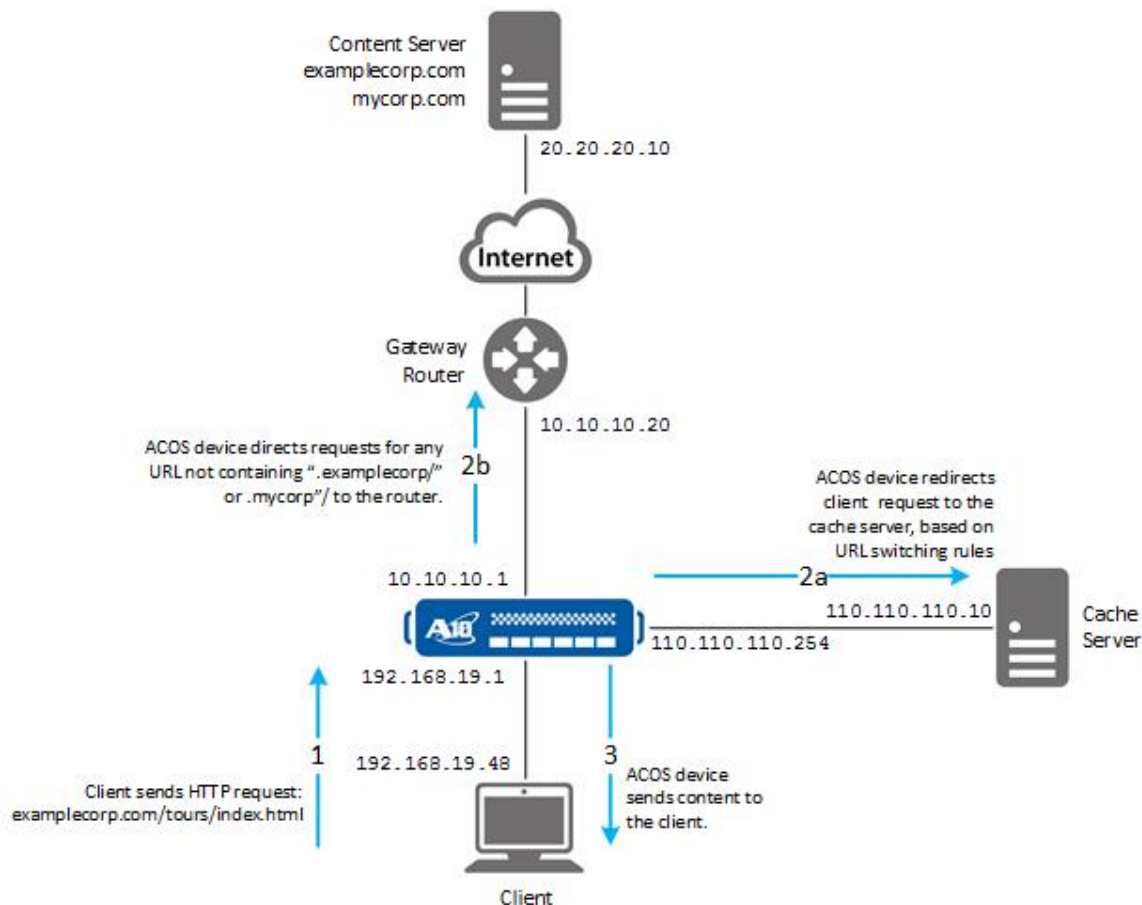
```
ACOS(config)# slb virtual-server wildcard 0.0.0.0 acl 198
ACOS(config-slb vserver)# port 80 http
ACOS(config-slb vserver-vport)# service-group sg-tcs
ACOS(config-slb vserver-vport)# no-dest-nat
ACOS(config-slb vserver-vport)# exit
ACOS(config-slb vserver)# exit
```

Service Type HTTP with URL Switching Rules

The service type HTTP with URL switching rules method uses an HTTP template containing URL switching rules. Traffic matching a URL switching rule is redirected to the cache server. Other traffic is sent to the gateway router. This method requires of a separate real server and service group for the gateway router.

[Figure 68](#) shows an example of the second method.

Figure 68 : Layer 7 TCS Using URL Switching Rules



Service Type HTTP With URL Switching Rules (Procedure)

1. Configure the interfaces connected to the clients, the content servers, and the cache server. Enable promiscuous VIP on the ACOS interface(s) connected to the clients.
2. Configure an extended ACL that uses the permit action and that matches on client addresses as the source address, and on the content server address as the destination address.
3. Configure a real server for the cache server. Add the TCP or UDP port; for example, TCP port 80.

4. Configure a real server for the next-hop router through which the ACOS device will reach the content servers. Add the same TCP port number as the one on the cache server (for example, TCP port 80). Disable health checking on the port.
5. The configuration requires health checking to be disabled on the router port. The router will not respond to the health check. If you leave health checking enabled, the ACOS device will mark the port down and TCS will not work.
6. Configure a service group for the cache server and add the cache server to it.
7. Configure a separate service group for the router, and add the router to it.
8. Configure an HTTP template with URL switching rules. Add a separate URL switching rule for each URI string based on which to select a service group.
9. Configure a virtual server with virtual IP address 0.0.0.0 (the wildcard VIP address) and bind it to the ACL.
10. Add virtual port 80 with service type HTTP and bind it to the service group containing the cache server. Bind the virtual port to the HTTP template. Enable/disable destination NAT.
11. Add virtual port 0 with service type HTTP and bind it to the service group containing the router. Enable/disable destination NAT.

Service Type HTTP Without URL Switching Rules (CLI Example)

The commands in this section implement the TCS configuration shown in [Figure 68](#). The commands for configuring the interfaces and ACL, and the real server and service group for the cache server, are the same as those used in the Layer 4 TCS example, and are therefore not shown.

The following commands configure a real server for the gateway router:

```
ACOS(config)# slb server router 10.10.10.20
ACOS(config-real server)# port 80 tcp
ACOS(config-real server-node port)# health-check-disable
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
```

The following commands configure a service group for the router:

```
ACOS(config)# slb service-group sg-router tcp
ACOS(config-slb svc group)# member router 80
ACOS(config-slb svc group-member:80)# exit
```

```
ACOS(config-slb svc group)# exit
```

The following commands configure an HTTP template containing URL switching rules. Client requests for any URL that contains “.examplecorp/” or “.mycorp/” will be redirected to the service group for the cache server. Requests for any other URL will instead be sent to the service group for the router.

```
ACOS(config)# slb template http http1  
ACOS(config-HTTP template)# url-switching contains .examplecorp/ service-  
group sg-tcs  
ACOS(config-HTTP template)# url-switching contains .mycorp/ service-group  
sg-tcs  
ACOS(config-HTTP template)# exit
```

The following commands configure a wildcard VIP and bind it to the ACL:

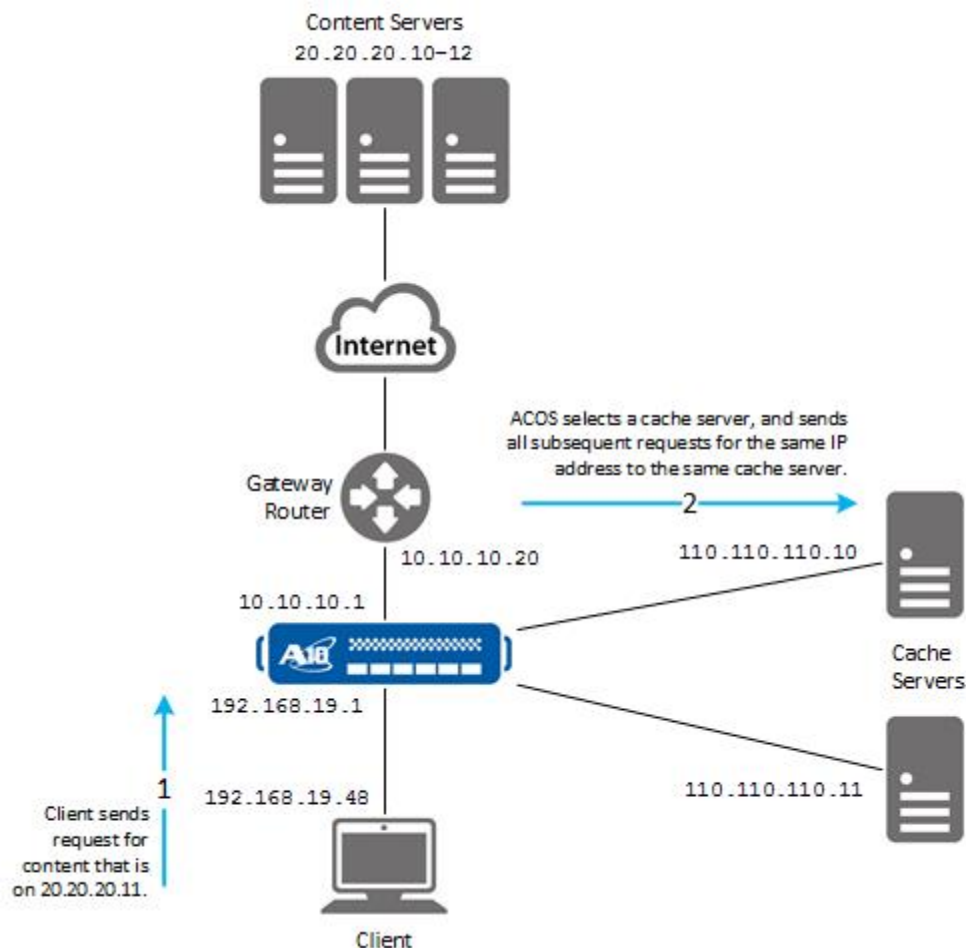
```
ACOS(config)# slb virtual-server wildcard 0.0.0.0 acl 198  
ACOS(config-slb vserver)# port 80 http  
ACOS(config-slb vserver-vport)# service-group sg-router  
ACOS(config-slb vserver-vport)# template http http1  
ACOS(config-slb vserver-vport)# no-dest-nat  
ACOS(config-slb vserver-vport)# exit  
ACOS(config-slb vserver)# exit
```

Optimizing TCS with Multiple Cache Servers

To optimize TCS in deployments that use multiple cache servers, use a destination-IP persistence template.

Commands in this section implement the TCS configuration shown in [Figure 69](#). Only commands specific to destination-IP persistence are shown. The other commands are the same as those in the previous sections.

Figure 69 : TCS with Multiple Cache Servers



Optimizing TCS with Multiple Cache Servers (CLI Example)

The following commands configure the destination-IP persistence template:

```
ACOS(config)# slb template persist destination-ip d-sticky
ACOS(config-dest ip persistence template)# match-type service-group
ACOS(config-dest ip persistence template)# exit
```

The **match-type service-group** command is required to enable URL switching and persistence in the same configuration.

The following commands configure the VIP. The commands are the same as those used for Layer 7 TCS, with the addition of a command to bind the destination-IP persistence template to the virtual port.


```
ACOS(config)# slb virtual-server wildcard 0.0.0.0 acl 198
ACOS(config-slb vserver)# port 80 http
ACOS(config-slb vserver-vport)# template http http1
ACOS(config-slb vserver-vport)# service-group sg-router
ACOS(config-slb vserver-vport)# no-dest-nat
ACOS(config-slb vserver-vport)# template persist destination-ip d-sticky
ACOS(config-slb vserver-vport)# exit
ACOS(config-slb vserver)# exit
```

Enabling Support for Cache Spoofing

If the cache server spoofs client IP addresses when requesting content from servers, the following additional configuration is required:

- Enable cache spoofing support on the ACOS interface connected to the spoofing cache server. Use the `ip cache-spoofing-port` command at the interface configuration level.
- In the real server configuration for the cache server, enable spoof caching support. Use the `spoofing-cache` command at the real server configuration level.

Enabling Support for Cache Spoofing (CLI Example)

The commands in this section enable cache spoofing support for the TCS configuration shown in [TCS with Multiple Cache Servers](#).

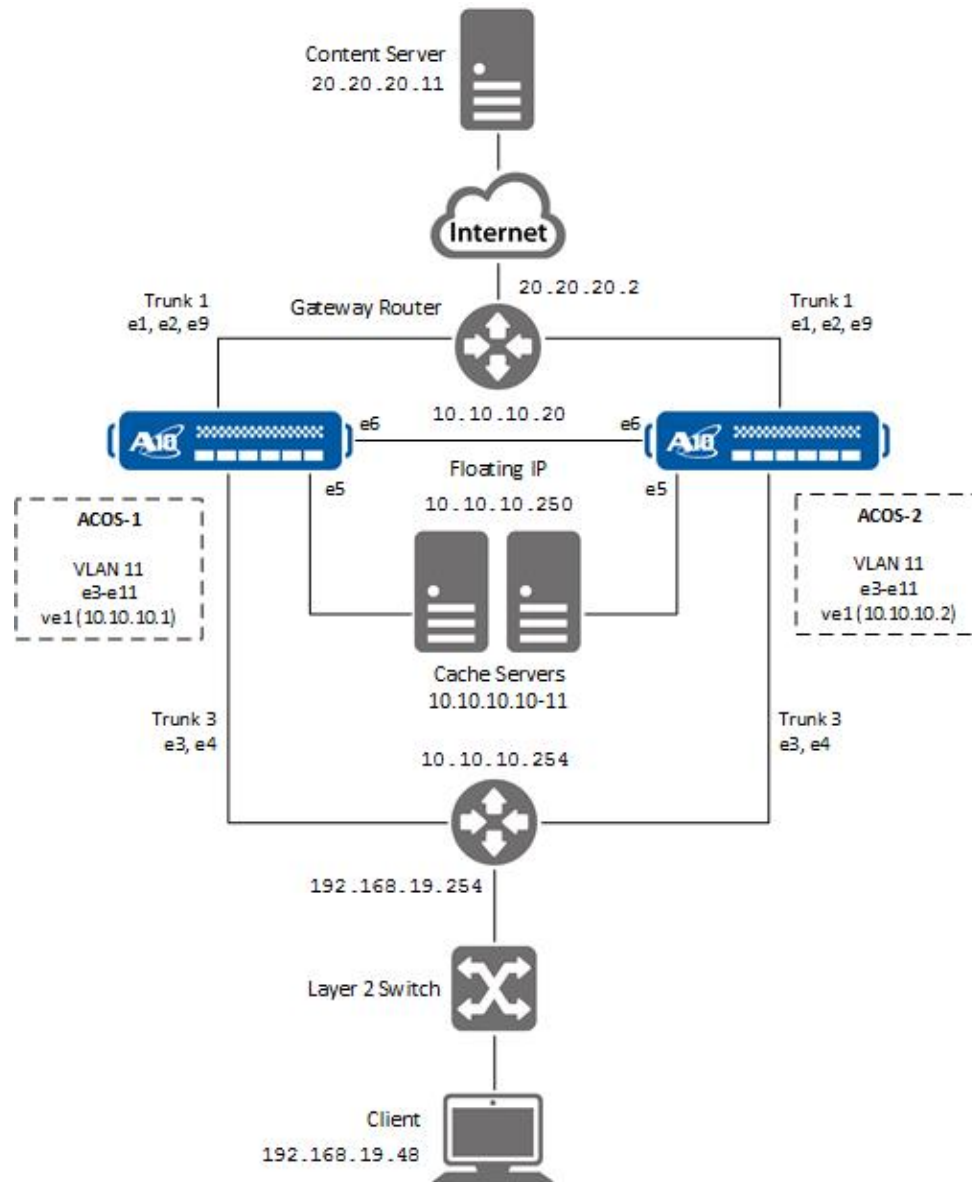
```
ACOS(config)# interface ethernet 5
ACOS(config-if:ethernet:5)# ip address 110.110.110.254 255.255.255.0
ACOS(config-if:ethernet:5)# ip cache-spoofing-port
ACOS(config-if:ethernet:5)# exit
ACOS(config)# slb server cache-rs 110.110.110.10
ACOS(config-real server)# spoofing-cache
ACOS(config-real server)# port 80 tcp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
```

Configuring TCS: VRRP-A High Availability

TCS: IPv4 VRRP-A High Availability Layer 3 Inline Mode

Use VRRP-A high availability to provide redundancy and failover for TCS. This section shows an example for IPv4 Layer 3 inline mode VRRP-A high availability. Layer 3 high availability for inline mode is beneficial in network topologies where the ACOS interfaces with the clients and cache servers are in the same subnet. [Figure 70](#) shows an example.

Figure 70 : TCS in VRRP-A high availability Layer 3 Inline Mode



Configuring IPv4 TCS in VRRP-A High Availability Layer 3 Inline Mode (Procedure)

Interface Parameters

Each ACOS device connects to the client, cache servers, and content server on a single IP interface:

- ACOS-1 – Connected on IP interface 10.10.10.1, assigned to VE 1 on VLAN 1, Ethernet data ports 3-11
- ACOS-2 – Connected on IP interface 10.10.10.2, assigned to VE 1 on VLAN 1, Ethernet data ports 3-11

The following interface parameters are required:

- Promiscuous VIP – Enable the interface connected to clients and the IP interface assigned to the VE on the VLAN containing the interfaces to the clients, content servers, and cache servers.
- Cache spoofing – If the cache server will spoof client IP addresses when requesting content from content servers, enable this feature on the ACOS interface connected to the cache server.

VRRP-A High Availability Parameters

- Device ID – ACOS-1 uses Device ID 1. ACOS-2 uses Device ID 2.
- VRID and priority – Configure a single VRID with a higher priority on ACOS-1.
- Pre-emption – Enable to force an initial failover to the ACOS device with higher priority.
- VRRP-A interfaces – Configure ethernet interfaces 1, 3, and 6. Interfaces 1 and 3 are the lead interfaces in trunks, so all the interfaces in these trunks are VRRP-A interfaces.
- Session synchronization (connection mirroring) – Enable to synchronize each ACOS device and its sessions onto the other ACOS device when in an active role.
- Floating IP address – Both ACOS devices share floating IP address 10.10.10.250 for the VRID.
- L3-inline mode – Enable on each ACOS device (specifically inline mode).
- Restart port list – Configure interfaces 1 to 5 and interface 9 are inline-mode restart ports. This includes the ACOS interfaces with the client, cache servers, and content server. Interface 6 is the dedicated HA link between the ACOS devices and is *not* included in the restart list (specifically inline mode).

NOTE: You must omit at least one port connecting the ACOS devices from the restart port-list, and heartbeat messages must be enabled on the port. This is to ensure that heartbeat messages between the ACOS devices are maintained; otherwise, flapping might occur.

SLB Parameters – Real server parameters

- Port type – Use Layer 4 port type, such as TCP. HA session synchronization is supported only for Layer 4 sessions.
- Cache spoofing – If the cache server will spoof client IP addresses when requesting content from content servers, enable this feature on the real server.

SLB Parameters – Service group parameters

- Type – Typically this should be TCP.
- Members – Add the real servers configured for the cache servers.

If the Layer 7 TCS configuration uses URL switching, a separate real server is required for the gateway router, and the real server is required to be placed in its own service group. (See [Configuring Layer 7 TCS](#).) The example in [Figure 70](#) does not use Layer 7 switching.

SLB Parameters – Virtual server parameters

- VIP – Add the wildcard VIP address (0.0.0.0). The VIP must be associated with an extended ACL using permit action. This must match the client addresses as the source address and as the destination address on the content server.
- Service type – Add the service type of Layer 4 TCS virtual port (type TCP).
- VRID – Add the virtual server to the VRID.
- Destination NAT – Disable this feature.
- Session synchronization – Enable this feature with the `ha-conn-mirror` command so that the customer sessions are synchronized from the active ACOS device to the standby ACOS device. Use the `on-syn` command to synchronize sessions when the virtual port receives an SYN packet.

If the spoof-caching is enabled, the ACOS device might create a transparent session from the cache to the server to fetch data. This transparent session and the main data session from the client to the cache server are always

synchronized to the standby device when the service type is Layer 4 virtual port (TCP).

Templates

For simplicity, the sample configuration in this section does not use any custom templates. For information about the templates that can be used with TCS, see [Application Templates](#).

Configuring IPv4 TCS in VRRP-A High Availability Layer 3 Inline Mode (CLI Example)

The following CLI commands implement the configuration.

ACOS-1 Configuration

The following commands configure the links:

```
ACOS-1(config)# interface ethernet 1
ACOS-1(config-if:ethernet:1)# enable
ACOS-1(config-if:ethernet:1)# trunk-group 1
ACOS-1(config-if:ethernet:1-trunk-group:1)# exit
ACOS-1(config-if:ethernet:1)# exit
ACOS-1(config)# interface ethernet 2
ACOS-1(config-if:ethernet:2)# enable
ACOS-1(config-if:ethernet:2)# trunk-group 1
ACOS-1(config-if:ethernet:2-trunk-group:1)# exit
ACOS-1(config-if:ethernet:2)# exit
ACOS-1(config)# interface ethernet 9
ACOS-1(config-if:ethernet:9)# enable
ACOS-1(config-if:ethernet:9)# trunk-group 1
ACOS-1(config-if:ethernet:9-trunk-group:1)# exit
ACOS-1(config-if:ethernet:9)# exit
ACOS-1(config)# interface ethernet 3
ACOS-1(config-if:ethernet:3)# enable
ACOS-1(config-if:ethernet:3)# ip allow-promiscuous-vip
ACOS-1(config-if:ethernet:3)# trunk-group 3
ACOS-1(config-if:ethernet:3-trunk-group:3)# exit
ACOS-1(config-if:ethernet:3)# exit
ACOS-1(config)# interface ethernet 4
ACOS-1(config-if:ethernet:4)# enable
ACOS-1(config-if:ethernet:4)# trunk-group 3
```

```

ACOS-1(config-if:ethernet:4-trunk-group:3)# exit
ACOS-1(config-if:ethernet:4)# exit
ACOS-1(config)# vlan 11
ACOS-1(config-vlan:11)# untagged trunk 3
ACOS-1(config-vlan:11)# tagged trunk 1
ACOS-1(config-vlan:11)# router-interface ve 11
ACOS-1(config-vlan:11)# exit
ACOS-1(config)# interface ethernet 5
ACOS-1(config-if:ethernet:5)# enable
ACOS-1(config-if:ethernet:5)# ip cache-spoofing-port
ACOS-1(config-if:ethernet:5)# exit
ACOS-1(config)# interface ve 11
ACOS-1(config-if:ve11)# ip address 10.10.10.1 255.255.255.0
ACOS-1(config-if:ve11)# ip allow-promiscuous-vip
ACOS-1(config-if:ve11)# exit

```

These commands configure static routes. One route goes to the subnet on the other side of the router connecting the ACOS device to the content servers. The other static route goes to the subnet on the other side of the router that connects the ACOS device to the client.

```

ACOS-1(config)# ip route 20.20.20.0 /24 10.10.10.20
ACOS-1(config)# ip route 192.168.19.0 /24 10.10.10.254

```

The following command configures an extended ACL that uses the permit action and that matches on client addresses as the source address, and on the content server address as the destination address:

```

ACOS-1(config)# access-list 198 permit ip any host 20.20.20.11 log

```

The following commands configure the global VRRP-A parameters:

```

ACOS-1(config)# vrrp-a common
ACOS-1(config-common)# device-id 1
ACOS-1(config-common)# set-id 1
ACOS-1(config-common)# enable
ACOS-1(config-common)# disable-default-vrid
ACOS-1(config-common)# exit
ACOS-1(config)# vrrp-a l3-inline-mode
ACOS-1(config)# vrrp-a vrid 1
ACOS-1(config-vrid:1)# floating-ip 10.10.10.250
ACOS-1(config-vrid:1)# blade-parameters
ACOS-1(config-vrid:1-blade-parameters)# priority 200

```

```
ACOS-1(config-vrid:1-blade-parameters)# exit
ACOS-1(config-vrid:1)# exit
ACOS-1(config)# vrrp-a interface ethernet 6
ACOS-1(config-ethernet:6)# vlan 11
ACOS-1(config-ethernet:6)# exit
ACOS-1(config)# vrrp-a restart-port-list
ACOS-1(config-restart-port-list)# ethernet 1 to 5
ACOS-1(config-restart-port-list)# ethernet 9
ACOS-1(config-restart-port-list)# exit
```

The following commands configure real servers for the cache servers:

```
ACOS-1(config)# slb server cache1 10.10.10.10
ACOS-1(config-real server)# spoofing-cache
ACOS-1(config-real server)# port 80 tcp
ACOS-1(config-real server-node port)# exit
ACOS-1(config-real server)# exit
ACOS-1(config)# slb server cache2 10.10.10.11
ACOS-1(config-real server)# spoofing-cache
ACOS-1(config-real server)# port 80 tcp
ACOS-1(config-real server-node port)# exit
ACOS-1(config-real server)# exit
```

The following commands configure a service group for the real servers:

```
ACOS-1(config)# slb service-group sg-cache-80 tcp
ACOS-1(config-slb svc group)# member cache1 80
ACOS-1(config-slb svc group-member:80)# exit
ACOS-1(config-slb svc group)# member cache2 80
ACOS-1(config-slb svc group-member:80)# exit
```

The following commands configure the virtual server:

```
ACOS-1(config)# slb virtual-server wildcard 0.0.0.0 acl 198
ACOS-1(config-slb vserver)# vrid 1
ACOS-1(config-slb vserver)# port 80 tcp
ACOS-1(config-slb vserver-vport)# service-group sg-cache-80
ACOS-1(config-slb vserver-vport)# no-dest-nat
ACOS-1(config-slb vserver-vport)# ha-conn-mirror
ACOS-1(config-slb vserver-vport)# exit
ACOS-1(config-slb vserver)# exit
```


ACOS-2 Configuration

The commands on ACOS-2 are the same as the ones on ACOS-1, with the following exceptions:

The **ip address** command on VE adds a unique IP address (not the address of the other ACOS device).

The **vrid** command assigns VIRD 2 instead of VRID 1.

The **priority** command assigns a lower priority to the group.

```
ACOS-2(config)# interface ethernet 1
ACOS-2(config-if:ethernet:1)# enable
ACOS-2(config-if:ethernet:1)# trunk-group 1
ACOS-2(config-if:ethernet:1-trunk-group:1)# exit
ACOS-2(config-if:ethernet:1)# exit
ACOS-2(config)# interface ethernet 2
ACOS-2(config-if:ethernet:2)# enable
ACOS-2(config-if:ethernet:2)# trunk-group 1
ACOS-2(config-if:ethernet:2-trunk-group:1)# exit
ACOS-2(config-if:ethernet:2)# exit
ACOS-2(config)# interface ethernet 9
ACOS-2(config-if:ethernet:9)# enable
ACOS-2(config-if:ethernet:9)# trunk-group 1
ACOS-2(config-if:ethernet:9-trunk-group:1)# exit
ACOS-2(config-if:ethernet:9)# exit
ACOS-2(config)# interface ethernet 3
ACOS-2(config-if:ethernet:3)# enable
ACOS-2(config-if:ethernet:3)# ip allow-promiscuous-vip
ACOS-2(config-if:ethernet:3)# trunk-group 3
ACOS-2(config-if:ethernet:3-trunk-group:3)# exit
ACOS-2(config-if:ethernet:3)# exit
ACOS-2(config)# interface ethernet 4
ACOS-2(config-if:ethernet:4)# enable
ACOS-2(config-if:ethernet:4)# trunk-group 3
ACOS-2(config-if:ethernet:4-trunk-group:3)# exit
ACOS-2(config-if:ethernet:4)# exit
ACOS-2(config)# vlan 11
ACOS-2(config-vlan:11)# untagged ethernet 3 to 6
ACOS-2(config-vlan:11)# tagged ethernet 1 to 2
ACOS-2(config-vlan:11)# router-interface ve 11
```

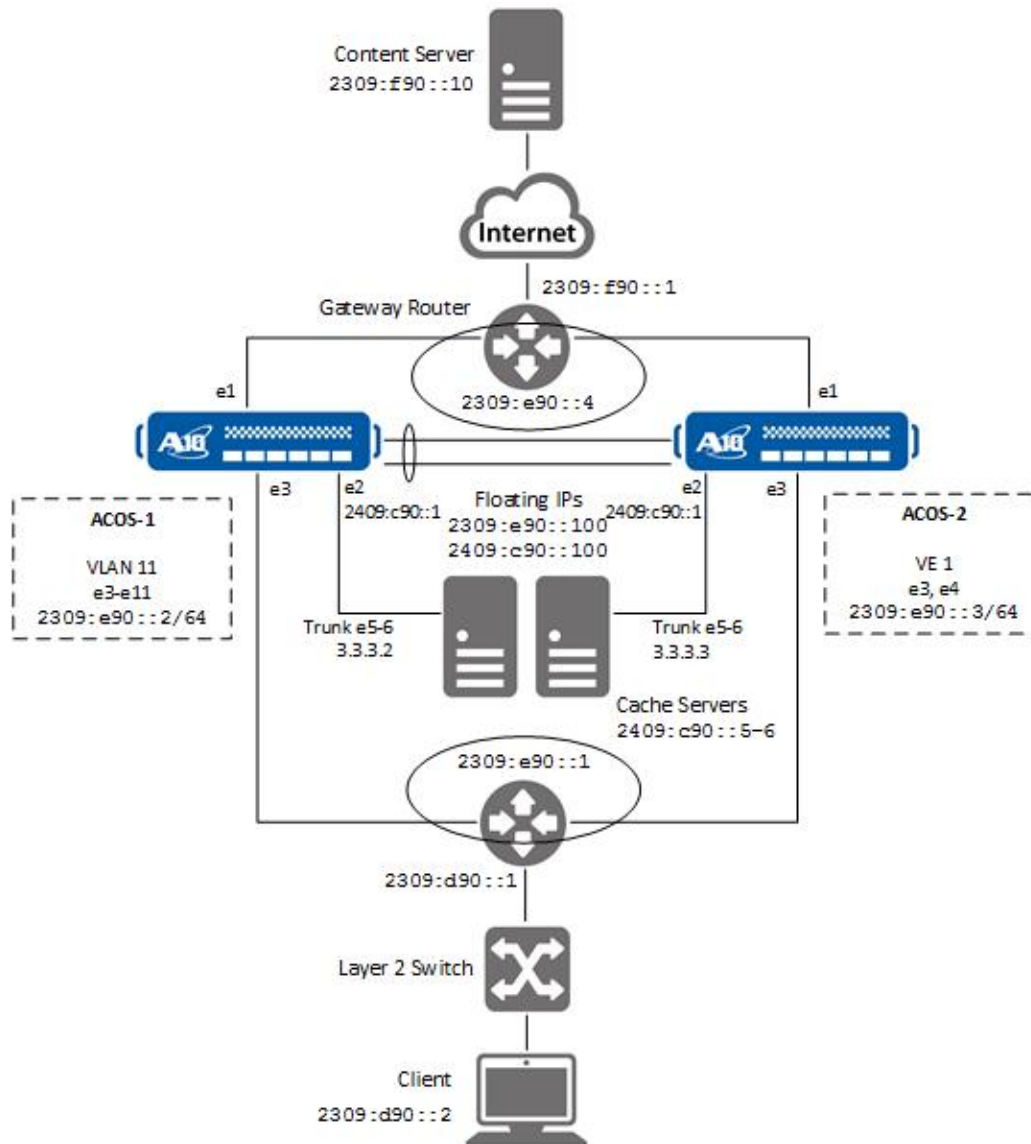
```
ACOS-2(config-vlan:11)# exit
ACOS-2(config)# interface ethernet 5
ACOS-2(config-if:ethernet:5)# enable
ACOS-2(config-if:ethernet:5)# ip cache-spoofing-port
ACOS-2(config-if:ethernet:5)# exit
ACOS-2(config)# interface ve 11
ACOS-2(config-if:ve11)# ip address 10.10.10.2 255.255.255.0
ACOS-2(config-if:ve11)# ip allow-promiscuous-vip
ACOS-2(config-if:ve11)# exit
ACOS-2(config)# ip route 20.20.20.0 /24 10.10.10.20
ACOS-2(config)# ip route 192.168.19.0 /24 10.10.10.254
ACOS-2(config)# access-list 198 permit ip any host 20.20.20.11 log
ACOS-2(config)# vrrp-a common
ACOS-2(config-common)# device-id 1
ACOS-2(config-common)# set-id 1
ACOS-2(config-common)# enable
ACOS-2(config-common)# disable-default-vrid
ACOS-2(config-common)# exit
ACOS-2(config)# vrrp-a l3-inline-mode
ACOS-2(config)# vrrp-a vrid 2
ACOS-2(config-vrid:1)# floating-ip 10.10.10.250
ACOS-2(config-vrid:1)# blade-parameters
ACOS-2(config-vrid:1-blade-parameters)# priority 180
ACOS-2(config-vrid:1-blade-parameters)# exit
ACOS-2(config-vrid:1)# exit
ACOS-2(config)# vrrp-a interface ethernet 6
ACOS-2(config-ethernet:6)# vlan 11
ACOS-2(config-ethernet:6)# exit
ACOS-2(config)# vrrp-a restart-port-list
ACOS-2(config-restart-port-list)# ethernet 1 to 5
ACOS-2(config-restart-port-list)# ethernet 9
ACOS-2(config-restart-port-list)# exit
ACOS-2(config)# slb server cache1 10.10.10.10
ACOS-2(config-real server)# spoofing-cache
ACOS-2(config-real server)# port 80 tcp
ACOS-2(config-real server-node port)# exit
ACOS-2(config-real server)# exit
ACOS-2(config)# slb server cache2 10.10.10.11
ACOS-2(config-real server)# spoofing-cache
ACOS-2(config-real server)# port 80 tcp
```

```
ACOS-2(config-real server-node port)# exit
ACOS-2(config-real server)# exit
ACOS-2(config)# slb service-group sg-cache-80 tcp
ACOS-2(config-slb svc group)# member cache1 80
ACOS-2(config-slb svc group)# member cache2 80
ACOS-2(config-slb svc group)# exit
ACOS-2(config)# slb virtual-server wildcard 0.0.0.0 acl 198
ACOS-2(config-slb vserver)# vrid 2
ACOS-2(config-slb vserver)# port 80 tcp
ACOS-2(config-slb vserver-vport)# service-group sg-cache-80
ACOS-2(config-slb vserver-vport)# no-dest-nat
ACOS-2(config-slb vserver-vport)# ha-conn-mirror
```

TCS : IPv6 VRRP-A High Availability Layer 3 Inline Mode

The configuration requirements and syntax are the same as for IPv4, differing only with using IPv6 addresses instead of IPv4 addresses. [Figure 71](#) shows an example of a TCS deployment in VRRP-A Layer 3 Inline mode.

Figure 71 : TCS – VRRP-A Layer 3 Inline Mode



Configuring IPv6 TCS in VRRP-A High Availability Layer 3 Inline Mode (CLI Example)

The following CLI commands implement the configuration.

ACOS-1 Configuration

The following commands configure the links.

```
ACOS-1(config)# interface ethernet 5
```

```
ACOS-1(config-if:ethernet:5)# enable
ACOS-1(config-if:ethernet:5)# trunk-group 1
ACOS-1(config-if:ethernet:5-trunk-group:1)# exit
ACOS-1(config-if:ethernet:5)# exit
ACOS-1(config)# interface ethernet 6
ACOS-1(config-if:ethernet:6)# enable
ACOS-1(config-if:ethernet:6)# trunk-group 1
ACOS-1(config-if:ethernet:6-trunk-group:1)# exit
ACOS-1(config-if:ethernet:6)# exit
ACOS-1(config)# vlan 21
ACOS-1(config-vlan:21)# untagged ethernet 1
ACOS-1(config-vlan:21)# router-interface ve 21
ACOS-1(config-vlan:21)# exit
ACOS-1(config)# vlan 22
ACOS-1(config-vlan:22)# untagged ethernet 2
ACOS-1(config-vlan:22)# router-interface ve 22
ACOS-1(config-vlan:22)# exit
ACOS-1(config)# vlan 56
ACOS-1(config-vlan:56)# untagged ethernet 5 to 6
ACOS-1(config-vlan:56)# router-interface ve 56
ACOS-1(config-vlan:56)# exit
ACOS-1(config)# interface ethernet 2
ACOS-1(config-if:ethernet:2)# ip cache-spoofing-port
ACOS-1(config-if:ethernet:2)# exit
ACOS-1(config)# interface ve 21
ACOS-1(config-if:ve21)# ipv6 address 2309:e90::2/64
ACOS-1(config-if:ve21)# ip allow-promiscuous-vip
ACOS-1(config-if:ve21)# exit
ACOS-1(config)# interface ve 22
ACOS-1(config-if:ve22)# ipv6 address 2409:c90::1/64
ACOS-1(config-if:ve22)# exit
ACOS-1(config)# interface ve 56
ACOS-1(config-if:ve56)# ipv6 address 2509:c90::1/64
ACOS-1(config-if:ve56)# ip address 3.3.3.2 255.255.255.0
ACOS-1(config-if:ve56)# exit
```

These commands configure static routes. One of the routes goes to the subnet on the other side of the router that connects the ACOS device to the content servers. The other static route goes to the subnet on the other side of the router that connects the ACOS device to the client. CPU processing is also enabled on the routes.

```
ACOS-1(config)# ipv6 route 2309:d90::/32 2309:e90::1  
ACOS-1(config)# ipv6 route 2309:f90::/32 2309:e90::3
```

The following commands configure an IPv6 ACL that uses the permit action and that matches on client addresses as the source address, and on the content server address as the destination address:

```
ACOS-1(config)# ipv6 access-list ipv6-101  
ACOS-1(config-access-list:ipv6-101)# permit ipv6 any host 2309:f90::10 log  
ACOS-1(config-access-list:ipv6-101)# exit
```

The following commands configure the VRRP-A parameters:

```
ACOS-1(config)# vrrp-a common  
ACOS-1(config-common)# set-id 1  
ACOS-1(config-common)# device-id 1  
ACOS-1(config-common)# enable  
ACOS-1(config-common)# disable-default-vrid  
ACOS-1(config-common)# exit  
ACOS-1(config)# vrrp-a l3-inline-mode  
ACOS-1(config)# vrrp-a vrid 2  
ACOS-1(config-vrid:1)# floating-ip 2409:c90::100  
ACOS-1(config-vrid:1)# floating-ip 2309:e90::100  
ACOS-1(config-vrid:1)# blade-parameters  
ACOS-1(config-vrid:1-blade-parameters)# priority 200  
ACOS-1(config-vrid:1-blade-parameters)# exit  
ACOS-1(config-vrid:1)# exit  
ACOS-1(config)# vrrp-a interface ethernet 1  
ACOS-1(config-ethernet:1)# server-interface  
ACOS-1(config-ethernet:1)# no-heartbeat  
ACOS-1(config-ethernet:1)# exit  
ACOS-1(config)# vrrp-a interface ethernet 3  
ACOS-1(config-ethernet:1)# router-interface  
ACOS-1(config-ethernet:1)# no-heartbeat  
ACOS-1(config-ethernet:1)# exit  
ACOS-1(config)# vrrp-a restart-port-list  
ACOS-1(config-restart-port-list)# ethernet 1  
ACOS-1(config-restart-port-list)# ethernet 3  
ACOS-1(config-restart-port-list)# exit
```

These commands configure a custom ICMP health monitor with short interval and timeout values. In Layer 3 inline VRRP-A configurations, short interval and timeout values reduce traffic disruption after a failover.

```
ACOS-1(config)# health monitor icmp
ACOS-1(config-health:monitor)# interval 1 timeout 1
ACOS-1(config-health:monitor)# exit
```

The following commands configure ICMP health checking for the upstream and downstream routers. The health checks help ensure rapid VRRP-A failover. (See the *Configuring VRRP-A High Availability* guide.) The custom ICMP health monitor configured above is also used.

```
ACOS-1(config)# slb server up-router 2309:e90::1
ACOS-1(config-real server)# health-check icmp
ACOS-1(config-real server)# exit
ACOS-1(config)# slb server down-router 2309:e90::3
ACOS-1(config-real server)# health-check icmp
ACOS-1(config-real server)# exit
```

The following commands configure real servers for the cache servers:

```
ACOS-1(config)# slb server cache1-ipv6 2409:c90::5
ACOS-1(config-real server)# spoofing-cache
ACOS-1(config-real server)# health-check icmp
ACOS-1(config-real server)# port 80 tcp
ACOS-1(config-real server-node port)# exit
ACOS-1(config-real server)# exit
ACOS-1(config)# slb server cache2-ipv6 2409:c90::6
ACOS-1(config-real server)# spoofing-cache
ACOS-1(config-real server)# health-check icmp
ACOS-1(config-real server)# port 80 tcp
ACOS-1(config-real server-node port)# exit
ACOS-1(config-real server)# exit
```

The following commands configure a service group for the real servers (cache servers):

```
ACOS-1(config)# slb service-group cache-ipv6 tcp
ACOS-1(config-slb svc group)# member cache1-ipv6 80
ACOS-1(config-slb svc group-member:80)# member cache2-ipv6 80
ACOS-1(config-slb svc group-member:80)# exit
ACOS-1(config-slb svc group)# exit
```

The following commands configure the virtual server:

```
ACOS-1(config)# slb virtual-server wildcard-ipv6 :: ipv6-acl ipv6-101
ACOS-1(config-slb vserver)# vrid 1
ACOS-1(config-slb vserver)# port 80 tcp
ACOS-1(config-slb vserver-vport)# service-group cache-ipv6
ACOS-1(config-slb vserver-vport)# no-dest-nat
ACOS-1(config-slb vserver-vport)# ha-conn-mirror
ACOS-1(config-slb vserver-vport)# exit
ACOS-1(config-slb vserver)# exit
```

ACOS-2 Configuration

Here are the configuration commands for ACOS-2. Most of the commands are exactly the same as on ACOS-1. Only the following values differ:

- IP addresses of the VEs
- VRID priority

IP address for session synchronization (`ha conn-mirror`)

```
ACOS-2(config)# interface ethernet 5
ACOS-2(config-if:ethernet:5)# enable
ACOS-2(config-if:ethernet:5)# trunk-group 1
ACOS-2(config-if:ethernet:5-trunk-group:1)# exit
ACOS-2(config-if:ethernet:5)# exit
ACOS-2(config)# interface ethernet 6
ACOS-2(config-if:ethernet:6)# enable
ACOS-2(config-if:ethernet:6)# trunk-group 1
ACOS-2(config-if:ethernet:6-trunk-group:1)# exit
ACOS-2(config-if:ethernet:6)# exit
ACOS-2(config)# vlan 21
ACOS-2(config-vlan:21)# untagged ethernet 1 to 3
ACOS-2(config-vlan:21)# router-interface ve 21
ACOS-2(config-vlan:21)# exit
ACOS-2(config)# vlan 22
ACOS-2(config-vlan:22)# untagged ethernet 2
ACOS-2(config-vlan:22)# router-interface ve 22
ACOS-2(config-vlan:22)# exit
ACOS-2(config)# vlan 56
ACOS-2(config-vlan:56)# untagged ethernet 5 to 6
```



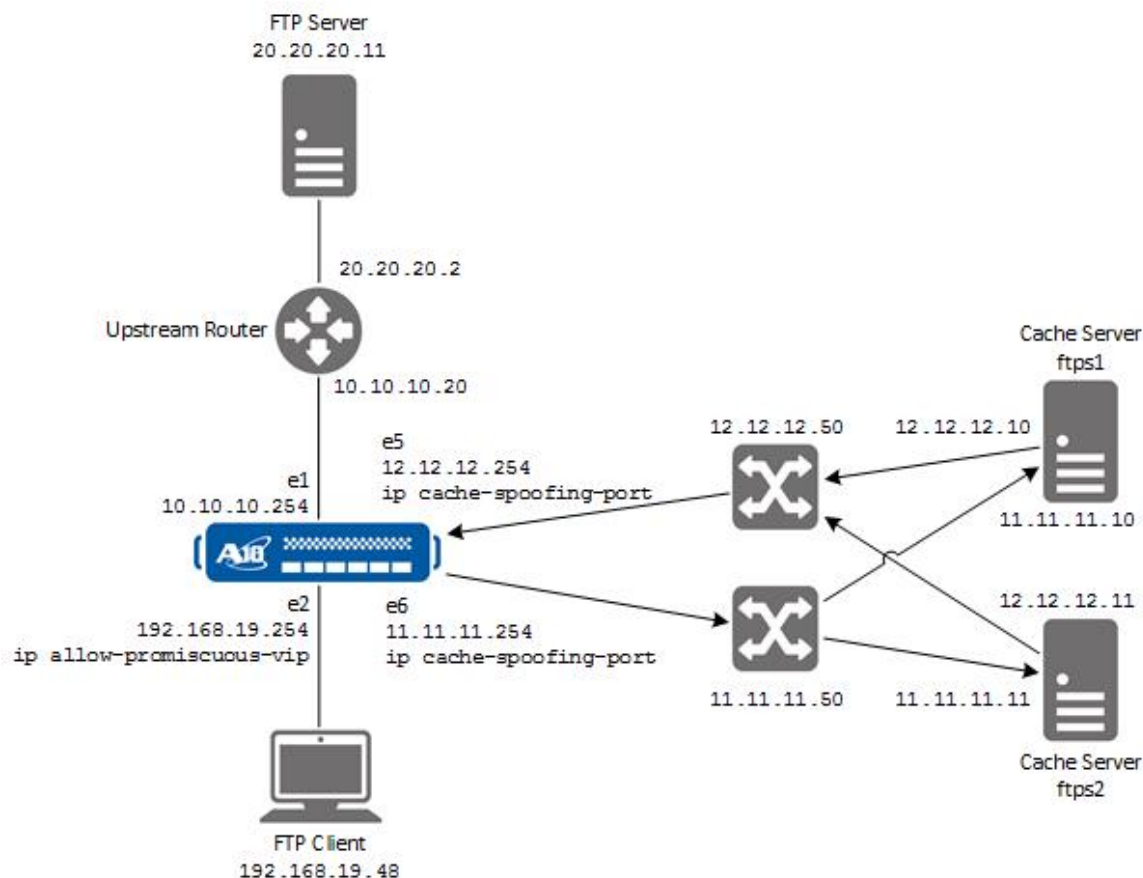
```
ACOS-2(config-vlan:56)# router-interface ve 56
ACOS-2(config-vlan:56)# exit
ACOS-2(config)# interface ethernet 2
ACOS-2(config-if:ethernet:2)# ip cache-spoofing-port
ACOS-2(config-if:ethernet:2)# exit
ACOS-2(config)# interface ve 21
ACOS-2(config-if:ve21)# ipv6 address 2309:e90::3/64
ACOS-2(config-if:ve21)# ip allow-promiscuous-vip
ACOS-2(config-if:ve21)# exit
ACOS-2(config)# interface ve 22
ACOS-2(config-if:ve22)# ipv6 address 2409:c90::1/64
ACOS-2(config-if:ve22)# exit
ACOS-2(config)# interface ve 56
ACOS-2(config-if:ve56)# ipv6 address 2509:c90::1/64
ACOS-2(config-if:ve56)# ip address 3.3.3.2 255.255.255.0
ACOS-2(config-if:ve56)# exit
ACOS-2(config)# ipv6 route 2309:d90::/32 2309:e90::1
ACOS-2(config)# ipv6 route 2309:f90::/32 2309:e90::3
ACOS-2(config)# ipv6 access-list ipv6-101
ACOS-2(config-access-list:ipv6-101)# permit ipv6 any host 2309:f90::10 log
ACOS-2(config-access-list:ipv6-101)# exit
ACOS-2(config)# vrrp-a common
ACOS-2(config-common)# set-id 1
ACOS-2(config-common)# device-id 1
ACOS-2(config-common)# enable
ACOS-2(config-common)# disable-default-vrid
ACOS-2(config-common)# exit
ACOS-2(config)# vrrp-a l3-inline-mode
ACOS-2(config)# vrrp-a vrid 2
ACOS-2(config-vrid:1)# floating-ip 2409:c90::100
ACOS-2(config-vrid:1)# floating-ip 2309:e90::100
ACOS-2(config-vrid:1)# blade-parameters
ACOS-2(config-vrid:1-blade-parameters)# priority 180
ACOS-2(config-vrid:1-blade-parameters)# exit
ACOS-2(config-vrid:1)# exit
ACOS-2(config)# vrrp-a interface ethernet 1
ACOS-2(config-ethernet:1)# server-interface
ACOS-2(config-ethernet:1)# no-heartbeat
ACOS-2(config-ethernet:1)# exit
ACOS-2(config)# vrrp-a interface ethernet 3
```

```
ACOS-2(config-ethernet:1)# router-interface
ACOS-2(config-ethernet:1)# no-heartbeat
ACOS-2(config-ethernet:1)# exit
ACOS-2(config)# vrrp-a restart-port-list
ACOS-2(config-restart-port-list)# ethernet 1
ACOS-2(config-restart-port-list)# ethernet 3
ACOS-2(config-restart-port-list)# exit
ACOS-2(config)# health monitor icmp
ACOS-2(config-health:monitor)# interval 1 timeout 1
ACOS-2(config-health:monitor)# exit
ACOS-2(config)# health monitor icmp interval 1 timeout 1
ACOS-2(config)# slb server up-router 2309:e90::1
ACOS-2(config-real server)# health-check icmp
ACOS-2(config-real server)# exit
ACOS-2(config)# slb server down-router 2309:e90::3
ACOS-2(config-real server)# health-check icmp
ACOS-2(config-real server)# exit
ACOS-2(config)# slb server cache1-ipv6 2409:c90::5
ACOS-2(config-real server)# spoofing-cache
ACOS-2(config-real server)# health-check icmp
ACOS-2(config-real server)# port 80 tcp
ACOS-2(config-real server-node port)# exit
ACOS-2(config-real server)# exit
ACOS-2(config)# slb server cache2-ipv6 2409:c90::6
ACOS-2(config-real server)# spoofing-cache
ACOS-2(config-real server)# health-check icmp
ACOS-2(config-real server)# port 80 tcp
ACOS-2(config-real server-node port)# exit
ACOS-2(config-real server)# exit
ACOS-2(config)# slb service-group cache-ipv6 tcp
ACOS-2(config-slb svc group)# member cache1-ipv6 80
ACOS-2(config-slb svc group-member:80)# member cache2-ipv6 80
ACOS-2(config-slb svc group-member:80)# exit
ACOS-2(config)# slb virtual-server wilddcard-ipv6 :: ipv6-acl ipv6-101
ACOS-2(config-slb vserver)# vrid 1
ACOS-2(config-slb vserver)# port 80 tcp
ACOS-2(config-slb vserver-vport)# service-group cache-ipv6
ACOS-2(config-slb vserver-vport)# no-dest-nat
ACOS-2(config-slb vserver-vport)# ha-conn-mirror
```

Configuring TCS for FTP

You can configure the ACOS device to use cache servers for FTP traffic as shown in [Figure 72](#).

Figure 72 : Transparent Cache Switching for FTP



When a client sends a request to the FTP server, the ACOS device intercepts the request and forwards it to the FTP cache server. The cache server then forwards the requested content to the ACOS device, if the content is cached. ACOS forwards the content to the client.

If the requested content is not already cached, the cache server obtains the content from the FTP server and caches it. ACOS forwards the content to the client.

Each cache server in this example has two physical interfaces. One of the interfaces receives client requests forwarded by the ACOS device. The other interface

communicates with the FTP server, and forwards cached content to the ACOS device. Only the interfaces that receive client requests from the ACOS device need to be configured as real servers.

NOTE: In this example, the content transferred by FTP is cached on the cache servers. However, this feature also can be used if the device is a firewall instead of an FTP cache server. In that case, the firewall is used to examine the traffic that is transferred to or from the FTP server by the client.

Configuring TCS for FTP (Procedure)

1. Configure the interfaces connected to the clients, the content servers, and the cache server.
2. Enable promiscuous VIP on the ACOS interface(s) connected to the clients.
3. Enable cache spoofing on the interface(s) connected to the cache server.
4. Configure an extended ACL that uses the permit action and that matches on client addresses as the source address, and on the content server address as the destination address.
5. Configure a real server for the cache server. Add an FTP port to the server.
6. If the cache server will spoof client IP addresses when requesting content from content servers, enable cache spoofing support.
7. If the cache server has multiple interfaces, configure a separate real server for each one.
8. Configure a real server for the next-hop router through which the ACOS device will reach the content servers. Add the same FTP port number as the one on the cache server (for example, port 21). Disable health checking on the port.
9. The configuration requires health checking to be disabled on the router port. The router will not respond to the health check. If you leave health checking enabled, the ACOS device will mark the port down and TCS will not work.
10. Configure a service group for the cache servers and add them to it.
11. Configure a separate service group for the router, and add the router to it.

12. Configure a virtual server with virtual IP address 0.0.0.0 (the wildcard VIP address) and bind it to the ACL.
13. Add an FTP virtual port and bind it to the service group containing the cache server, and to the service group containing the router. Disable destination NAT on the virtual port.

Configuring TCS for FTP (CLI Example)

These commands configure the ACOS interfaces to the FTP server, the FTP client, and the cache servers.

```
ACOS(config)# interface ethernet 1
ACOS(config-if:ethernet:1)# enable
ACOS(config-if:ethernet:1)# ip address 10.10.10.254 255.255.255.0
ACOS(config-if:ethernet:1)# exit
ACOS(config)# interface ethernet 2
ACOS(config-if:ethernet:2)# enable
ACOS(config-if:ethernet:2)# ip address 192.168.19.254 255.255.255.0
ACOS(config-if:ethernet:2)# ip allow-promiscuous-vip
ACOS(config-if:ethernet:2)# exit
ACOS(config)# interface ethernet 5
ACOS(config-if:ethernet:5)# enable
ACOS(config-if:ethernet:5)# ip address 12.12.12.254 255.255.255.0
ACOS(config-if:ethernet:5)# ip cache-spoofing-port
ACOS(config-if:ethernet:5)# exit
ACOS(config)# interface ethernet 6
ACOS(config-if:ethernet:6)# enable
ACOS(config-if:ethernet:6)# ip address 11.11.11.254 255.255.255.0
ACOS(config-if:ethernet:6)# ip cache-spoofing-port
ACOS(config-if:ethernet:6)# exit
```

This command configures an extended ACL to match clients and the content server. The ACL in this example matches on any source address (client IP address) and on the destination IP address of the content server.

```
ACOS(config)# access-list 198 permit ip any host 20.20.20.11 log
```

The following commands configure real servers for FTP on each of the cache servers. Cache spoofing is enabled and TCP port 21 is added to each real server.

```
ACOS(config)# slb server ftps1 11.11.11.10
ACOS(config-real server)# spoofing-cache
```

```
ACOS(config-real server)# port 21 tcp
ACOS(config-real server-node port)# health-check-disable
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
ACOS(config)# slb server ftps2 11.11.11.11
ACOS(config-real server)# spoofing-cache
ACOS(config-real server)# port 21 tcp
ACOS(config-real server-node port)# health-check-disable
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
```

The following commands configure an FTP service group for the cache server:

```
ACOS(config)# slb service-group sg-ftp tcp
ACOS(config-slb svc group)# member ftps1 21
ACOS(config-slb svc group-member:21)# exit
ACOS(config-slb svc group)# member ftps2 21
ACOS(config-slb svc group-member:21)# exit
ACOS(config-slb svc group)# exit
```

The following commands configure a wildcard VIP traffic and bind it to the ACL. The FTP virtual port is bound to the FTP and router service groups. Also, destination NAT is disabled.

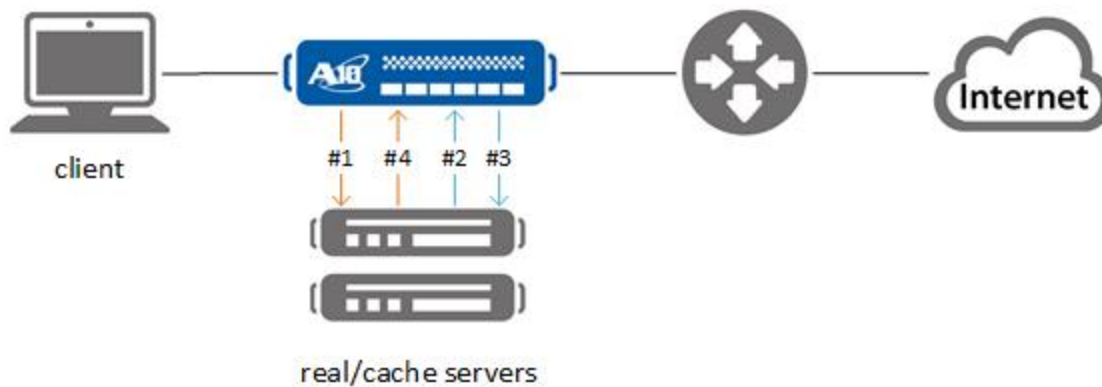
```
ACOS(config)# slb virtual-server wildcard 0.0.0.0 acl 198
ACOS(config-slb vserver)# port 21 ftp
ACOS(config-slb vserver-vport)# service-group sg-ftp
ACOS(config-slb vserver-vport)# no-dest-nat
ACOS(config-slb vserver-vport)# exit
ACOS(config-slb vserver)# exit
```

Configuring Bandwidth Limits Per Server or Port

You can create templates for monitoring and limiting the overall traffic load being handled by a real server or port. Once the threshold is reached, the server or port can avoid selection for newer sessions until the traffic load has subsided. It is also possible to enable accounting of traffic to and from the server, and logging if the traffic limits are exceeded.

This feature can be deployed for either Server Load Balancing (SLB) or Transparent Cache Switching (TCS) topology. In the TCS deployment considered in [Figure 73](#), there is SLB and TCS traffic flow:

Figure 73 : SLB and TCS Traffic Flow



1. To calculate the bandwidth for a real server or real port given both SLB traffic and TCS traffic flows through ACOS, the traffic rate is computed by counting the total bytes processed, corresponding to actual packets sent to and received from the real server (cache server) within a one second interval.
2. Client request packets are sent to the cache server (SLB session in orange).
3. Request packets are received from the cache server destined to the Internet (TCS session in blue).
4. Internet server response packets are sent to the cache server (TCS session in blue).
5. Response packets received from the cache server to be sent to the client (SLB session in orange).
6. Upon receiving a client request packet, ACOS will create an SLB session and then forward that packet to the cache server. The bytes in this request packet sent by ACOS to the cache server will count towards the traffic rate seen by the cache server.
7. Similarly when the cache server sends a request to the Internet server, ACOS will create a transparent session and subsequently such packets will be counted for the bandwidth computation.

8. Next when the Internet server response is received, it will be transmitted to the cache server and this packet length will count towards the cache server bandwidth computation.
9. Lastly when the cache server sends a response, the packet length will be counted towards the cache server bandwidth computation.
10. When the load exceeds the configured limit and duration interval, that cache server will no longer be included as part of the server selection process for newer flows. However, existing sessions continue to be processed and forwarded to the cache server.
11. If a persist template is configured with `dont-honor-conn-rules`, the real server/port will continue to be selected for new sessions, regardless of the threshold limit.

Configuring Bandwidth Rate Limit for Servers and Ports (Procedure and CLI Example)

To configure the bandwidth rate limit for a server or port, configure the `bw-rate-limit` in an SLB server or port template and apply the template to a real server or real port.

The following example shows how to configure the bandwidth rate limit of 1000 Kbps to exclude a real server from receiving new traffic flows when the threshold is exceeded for a duration of 5 seconds. The server will resume accepting new traffic flows after the bandwidth drops below 800 Kbps for a duration of 5 seconds. The rate limit messages will not be logged.

```
ACOS(config)# slb template server rate-limit-1
ACOS(config-rserver)# bw-rate-limit 1000 resume 800 duration 5 no-logging
ACOS(config-rserver)# exit
```

To apply the template to a real server:

```
ACOS(config)# slb server rs1 10.10.1.1
ACOS(config-real server)# template server rate-limit-1
ACOS(config-real server)# exit
```

If the measured traffic rate is greater than the configured `bw-rate-limit` consistently for the specified `duration`, it will be considered in 'exceed' state. Once it is in 'exceed' state, the measured traffic rate needs to fall below the `resume` threshold consistently for the specified `duration` to be considered in 'resume' state. Exceed and resume

state transition is then logged (once per state change per real port or real server within a 60 second interval). Logging is enabled by default.

If this feature is enabled along with a feature, such as system resource template, that tracks bandwidth usage on a partition or resource and then takes action to drop packets exceeding the resource template threshold, inconsistent computation of the underlying bandwidth rate for traffic from the real server may result.

For more information, see “slb template server” in the Command Line Interface Reference.

Configuring Bandwidth Rate Limit Accounting (Procedure and CLI Example)

By default, all traffic sent by or received from real servers is factored in to bandwidth rate limit accounting. It is possible to configure the template so that only sent or received traffic is factored in to the accounting.

To configure the accounting for `bw-rate-limit`, use the `bw-rate-limit-acct` command. This command is only available for SLB server template configuration, and not for SLB port templates. Upon binding a server template under a real server with this option, all real ports under the real server will automatically be subject to the same accounting method.

The following example shows how to configure the bandwidth rate limit accounting only for traffic received from the real server.

```
ACOS(config)# slb template server s1  
ACOS(config-rserver)# bw-rate-limit-acct from-server-only
```

For more information, see “slb template server” in the ADC-CLI Command Line Interface Reference.

Configuring Destination Hash-Based TCS

ACOS provides a more optimal Transparent Cache Switching (TCS) solution that is based on hash of the destination IP address for persistence instead of session persistence based on destination IP address. With this feature, ACOS does not create a persistent session. Instead, ACOS device uses a hash based on the available number of servers that are in an “UP” state in the service group.

Destination persistence templates can be attached only to wildcard VIPs.

Configuring Hash-Based TCS (Example)

Three cache servers (cache server 1, cache server 2, and cache server 3) are configured with destination IP hash persistence and bound to a virtual port.

Cache server 1 temporarily goes down.

Traffic that was persisting to cache server 1 will be distributed between the remaining two cache servers, cache server 2 and cache server 3 (with a hash base of 2, since only 2 servers are operational and available to calculate persistence).

Traffic sent to cache server 2 and cache server 3 will continue to be sent to the servers. This traffic will not be recalculated. Only traffic that is configured to “persist” to cache server 1 gets recalculated. Redistribution is resumed and traffic is distributed among the total number of functional servers.

GUI Configuration

1. Navigate to **ADC >> Templates >> Persistence**.
2. Click **Create**, then select **Persist Destination IP** from the drop-down list.
3. Enter a name for the template.
4. Select the checkbox in the Hash Persist field.
5. Configure the other fields as desired; refer to the GUI online help for detailed information about each of the fields on this screen.
6. Click **OK**.

CLI Configuration

From the destination-ip persist template, configure TCS destination IP address hash persistence.

To ensure that the destination IP address hash calculation will persist even after a server fails, enter the following commands, where “dhash” is the name of the template that you are creating for destination IP hash persistence:

```
ACOS-Active(config)# slb template persist destination-ip dhash  
ACOS-Active(config-destination ip persist)# hash-persist  
ACOS-Active(config-destination ip persist)# exit
```

To view the existing configuration:

```
ACOS-Active(config)# show running | section dhash  
slb template persist destination-ip dhash  
    hash-persist
```

To remove IP address hash persistence, issue the following command:

```
ACOS-Active(config)# no slb template persist destination-ip dhash
```

Diameter AAA Load Balancing

This section describes SLB for the Diameter AAA protocol.

The following topics are covered:

Overview	737
Parameters	740
Configuration	744

Overview

Diameter is a successor to RADIUS that provides security and other enhancements not supported in RADIUS. Diameter load balancing enables the ACOS device to act as a proxy between Diameter clients and servers.

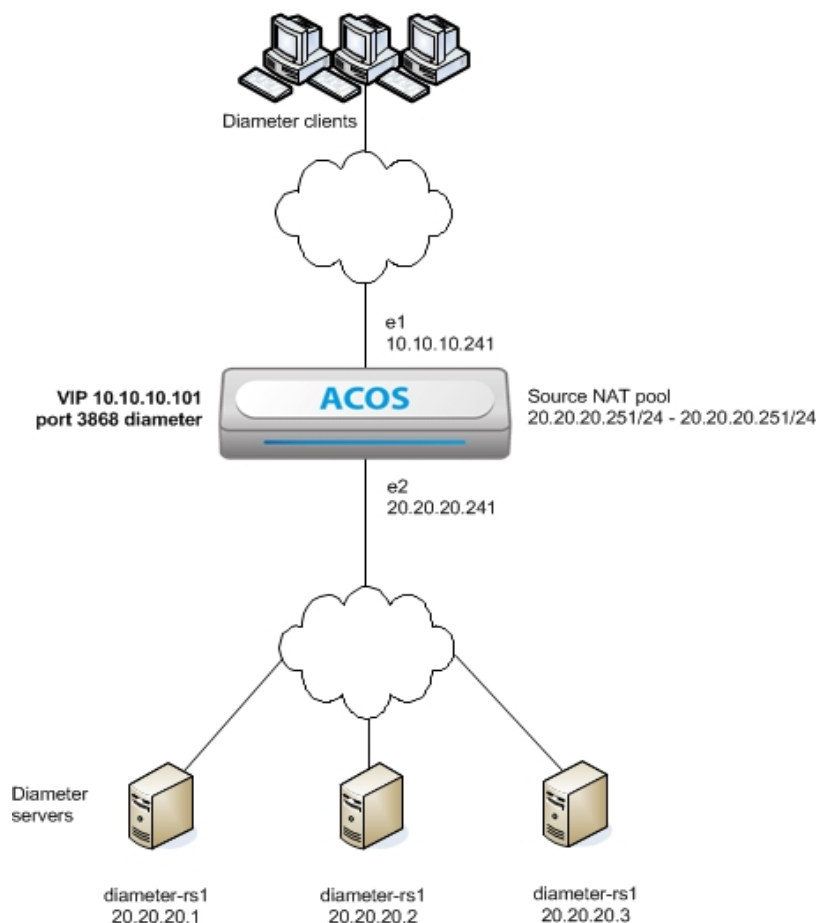
Diameter operates over TCP or SCTP. the ACOS device terminates the client's Diameter connection, and opens another Diameter connection with the selected server.

NOTE: The current release supports Diameter over TCP only.

Clients send Diameter messages, such as authentication requests, to the VIP configured on the ACOS device. ACOS uses SLB to select a Diameter server and forwards the client's request to the server. ACOS then forwards the server's reply to the client.

The clients and real servers can be connected to the ACOS device on Layer 2 or Layer 3. [Figure 74](#) shows an example.

Figure 74 : Diameter Load Balancing



Source NAT

Source NAT is required for traffic between the ACOS device and the Diameter servers. ACOS establishes a separate connection to each Diameter server before any client requests arrive. The NAT address(es), consisting of source IP address and source TCP port, uniquely identify the connections.

Load-Balancing

Diameter load balancing requires the strict round-robin load balancing method.

Session-ID persistence is automatically enabled for Diameter virtual ports. After a server is selected for a given client session, all requests for that session go to the same real server.

NOTE: You do not need to configure a session-ID persistence template. Session persistence is enabled automatically for Diameter virtual ports.

Optionally, you can configure multiple sets of service-group members (*server:port* pairs) that differ based on member priority. In this case, the ACOS device uses only the members with the highest priority as long as they are available, and uses lower-priority members only if the high-priority members become unavailable.

An additional option allows lower-priority members to temporarily be elevated to high priority to compensate for high-priority members that are unavailable.

Health Monitoring

You can use the Layer 3 health method (ICMP ping) to test the IP reachability of each server. Layer 3 health monitoring is enabled by default.

You do not need to configure any Layer 4 or Layer 7 health monitors on the ACOS device for Diameter load balancing. The Diameter protocol includes its own Layer 7 health method, the Device-Watchdog-Request message. ACOS periodically sends Device-Watchdog-Request messages to each Diameter real server. Each server must respond to its message within a configurable number of seconds or the server is marked Down.

NOTE: You will need to disable Layer 4 health monitoring, which is enabled by default. You can disable it individually in each server's configuration, or create a real port template for Diameter, disable the Layer 4 health monitor in the template, and assign the template to the TCP port in each real server configuration.

Application Templates

The following types of application templates are applicable to Diameter load balancing:

- TCP – Contains settings used for TCP connections between the ACOS device and Diameter clients and servers.
- Diameter – Contains the Diameter settings. (See [Parameters](#).)

Accounting Message Duplication

Optionally, you can configure the ACOS device to duplicate Accounting-Request messages and send them to a separate service group. This option is useful for logging, accounting, and so on.

Session-ID persistence is used to send all duplicate messages for a given client's session to the same server in the service group.

ACOS does not send Accounting-Answer messages in response to duplicate Accounting-Request messages sent to the duplication service group.

You can use a set of ACOS devices configured for High Availability (HA) to provide redundancy for Diameter load balancing. Make sure to enable session synchronization (also called “connection mirroring”) on the Diameter virtual port, to ensure that session-ID persistence is maintained across failovers.

NOTE:	The Diameter template parameter origin-host-id is not included in HA configuration synchronization. The other Diameter template parameters are included in HA configuration synchronization.

Parameters

The following parameters are configurable in Diameter templates.

Diameter Attribute-Value Pairs

You can configure the following Diameter Attribute-Value Pair (AVP) options:

- **Origin-host** – Sets the value of Diameter AVP 264. This AVP can be a character string and specifies the identity of the originating host for Diameter messages. Since the ACOS device acts as a proxy for Diameter, this AVP refers to the ACOS device itself, not to the actual clients. From the Diameter server's standpoint, the ACOS device is the Diameter client.

Specify the origin-host in the following format: *host.realm*

The *host* is a string unique to the client (ACOS device). The *realm* is the Diameter realm, specified by the origin-realm option (described below). There is no default.

- **Multiple-origin-host** – Prepends the ACOS CPU ID onto the origin-host string to identify the CPU used for a given Diameter peer connection. By default, this option is disabled and the CPU ID is not prepended onto the origin-host string.
- ACOS establishes a separate peer connection with each Diameter server on each CPU. The multiple-origin-host option does not enable or disable this behavior. The option simply shows or hides the CPU ID in the origin-host string.
- **Origin-realm** – Sets the value of Diameter AVP 296. This AVP can be a character string and specifies the Diameter realm from which Diameter messages, including requests, are originated. There is no default.
- **Product-name** – Sets the value of Diameter AVP 269. This AVP can be a character string and specifies the product; for example, "a10dra". There is no default.
- **terminate-on-cca-t** - Removes Diameter sessions when receiving the Server CCA-Termination message, rather than waiting for the Client Session-Terminate-Request (STR).
- **Vendor-ID** – Sets the value of Diameter AVP 266. This AVP can be a numeric value and specifies the vendor; for example, "156". Make sure to use a non-zero value. Zero is reserved by the Diameter protocol. There is no default.
- **AVP insertion** – Specifies custom AVP values to insert into Capabilities-Exchange-Request messages sent by the ACOS device to Diameter servers. For each custom AVP value to insert, you must specify the following information:
 - *avp-num* [mandatory] {int32 | int64 | string} *value*
 - *avp-num* – Diameter AVP number.
 - *mandatory* – Sets the AVP mandatory flag on. By default, this flag is off (not set).
 - *int32 | int64 | string* – Specifies the data format of the value to insert.

- *value* – Specifies the value to insert.
- You can configure up to 6 custom AVP values.
- Customize-CEA – Replaces the AVPs in Capabilities-Exchange-Answer (CEA) messages with the custom AVP values you configure before forwarding the messages. By default, this option is disabled.

Load Balancing for Specific Message Codes

By default, Diameter load balancing applies only to these message codes (also called “command codes”):

- Accounting-Request (code 271)
- Accounting-Answer (code 271)
- Credit-Control-Request (code 272)
- Credit-Control-Answer (code 272)
- Capabilities-Exchange-Request (code 257)
- Capabilities-Exchange-Answer (code 257)
- Device-Watchdog-Request (code 280)
- Device-Watchdog-Answer (code 280)
- Session-Termination-Request (code 275)
- Session-Termination-Answer (code 275)
- Abort-Session-Request (code 274)
- Abort-Session-Answer (code 274)
- Disconnect-Peer-Request/Disconnect-Peer-Answer (code 282)

ACOS drops all other Diameter message codes by default.

Optionally, use the message-code option to enable load balancing of additional Diameter message codes, on an individual message-code basis. You can enable load balancing of up to 10 additional message codes.

Timers

You can configure the following Diameter timers:

- Idle timeout – Specifies the number of minutes a Diameter session can remain idle before the session is deleted. You can specify 1-65535 minutes. The default is 5 minutes.
- Session-age – Specifies the absolute limit for Diameter sessions. Any Diameter session that is still in effect when the session age is reached is removed from the ACOS session table. You can specify 1-65535 minutes. The default is 10 minutes.
- DWR-time – Specifies the maximum number of seconds the ACOS device will wait for the reply to a device-watch-dog message sent to a Diameter server before marking the server Down. You can specify 0-2147483647 milliseconds (ms), in 100-ms increments. The default is 10 seconds.
- DWR-up-retry – Specifies the number of Device Watchdog Request and Device Watchdog Answer messages required to mark a server port as up. You can specify 1-7. The default is 3.

Accounting-Request Message Duplication

To configure message duplication, configure real servers and the service group, and configure the following option in the Diameter template:

Duplicate AVP – Specifies the Accounting-Request messages to duplicate, and the service group to which to send the duplicates. You must specify the following information:

- *avp-num pattern service-group*
- *avp-num* – Diameter AVP number.
- *pattern* – String pattern within the message.
- *service-group* – The duplication service group; the service group to send duplicate messages.

To place the message duplication configuration into effect, you must unbind the Diameter template from the Diameter virtual port, then rebind it.

A Diameter template where message duplication is configured can be bound to only one virtual port.

Configuration

To configure Diameter load balancing:

1. Configure a source NAT pool containing addresses in the same subnet as the Diameter servers.
2. Configure the real servers.
3. Configure the service group.
4. (Optional) Configure the real servers and service group for Accounting-Request message duplication.
5. Configure the Diameter template.
6. Configure the virtual server.

Diameter Load Balancing Configuration (CLI Procedure)

1. To configure the source NAT pool, use the `ip nat pool` command at the global configuration level of the CLI. For information about the command, see the *CLI Reference* or *System Configuration and Administration Guide*.
2. To configure the real servers, use the `slb server` command to change to the configuration level for the real server, where you can use the `port` command to add the TCP port to the server. The default Diameter protocol port is 3868.

The `port` command adds the port and changes the CLI to the configuration level for the port. At this level, use the `health-check-disable` command to disable the Layer 4 health monitor:

Instead, the ACOS device uses Diameter Device-Watchdog-Request messages to test the health of the Diameter protocol on the servers.

3. To configure the service group, use the `slb service-group` command.

This command changes the CLI to the configuration level for the service group, where you can use the `method round-robin-strict` command to add the real servers and service ports to the group:

The `member` command sets the load-balancing method required for Diameter load balancing. The command specifies the protocol port number of the service to be load balanced. Use the same port number specified in the server configuration in the above step. Repeat the command for each real server.

The `priority` option specifies the preference for using this server and port. You can specify 1-16. The highest priority value is 16 and the lowest is 1. The default is 1.

Normally, only the highest-priority members are used. Lower-priority members backups and are used only if the active members become unavailable. Optionally, you can use the `min-active-member` command to specify a minimum number of active members that are required. In this case, the ACOS device uses lower-priority servers even if some primary servers are still up.

4. To configure Accounting-Request message duplication, use the `slb server` and `slb service-group` commands to configure the real servers and service group. For information, see the following:

- [Accounting Message Duplication](#)
- [Parameters](#)

5. To configure the Diameter template, use the `slb template diameter` command.

This command changes the CLI to the configuration level for the template, where [Parameters](#) are available.

To configure the virtual server and virtual port, use the `slb virtual-server` command. This command changes the CLI to the configuration level for the virtual server, where you can use the `port` command to add the virtual port to the server.

The `port` command changes the CLI to the configuration level for the virtual port.

- Use the `source-nat pool` command to bind the virtual port to the source NAT pool.

- Use the **template diameter** command to bind the virtual port to the Diameter template.
 - Use the **service-group** command to bind the virtual port to the service group.
6. To verify and monitor Diameter load balancing operation, use the **show slb diameter** and **show slb server** commands.

Configuring Basic Diameter Load Balancing – Single ACOS Device (CLI Example)

This example configure the Diameter load-balancing deployment shown in [Diameter Load Balancing](#).

This command configures the source NAT pool:

```
ACOS(config)# ip nat pool snat.1 10.10.10.251 10.10.10.251 netmask /24
```

The following commands configure the real servers:

```
ACOS(config)# slb server diameter-rs1 10.10.10.1
ACOS(config-real server)# port 3868 tcp
ACOS(config-real server-node port)# health-check-disable
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
ACOS(config)# slb server diameter-rs2 10.10.10.2
ACOS(config-real server)# port 3868 tcp
ACOS(config-real server-node port)# health-check-disable
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
ACOS(config)# slb server diameter-rs3 10.10.10.3
ACOS(config-real server)# port 3868 tcp
ACOS(config-real server-node port)# health-check-disable
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
```

The following commands configure the service group. In this example, diameter-rs3 is a backup.

```
ACOS(config)# slb service-group sg-tcp3868 tcp
ACOS(config-slb svc group)# method round-robin-strict
ACOS(config-slb svc group)# member diameter-rs1 3868
ACOS(config-slb svc group-member:3868)# priority 1
ACOS(config-slb svc group-member:3868)# exit
ACOS(config-slb svc group)# member diameter-rs2 3868
```

```
ACOS(config-slb svc group-member:3868)# priority 1
ACOS(config-slb svc group-member:3868)# exit
ACOS(config-slb svc group)# member diameter-rs3 3868
ACOS(config-slb svc group-member:3868)# priority 2
ACOS(config-slb svc group-member:3868)# exit
ACOS(config-slb svc group)# exit
```

The following commands configure the Diameter template:

```
ACOS(config)# slb template diameter diameter1
ACOS(config-diameter template)# origin-host 2diameter.a10com
ACOS(config-diameter template)# origin-realm a10com
ACOS(config-diameter template)# product-name a10dra
ACOS(config-diameter template)# vendor-id 156
ACOS(config-diameter template)# exit
```

The following commands configure the VIP:

```
ACOS(config)# slb virtual-server vip-diameter.1 172.16.10.101
ACOS(config-slb vserver)# port 3868 diameter
ACOS(config-slb vserver-vport)# source-nat pool snat.1
ACOS(config-slb vserver-vport)# template diameter diameter1
ACOS(config-slb vserver-vport)# service-group sg-tcp3868
ACOS(config-slb vserver-vport)# exit
ACOS(config-slb vserver)# exit
```

Accounting-Request Message Duplication (CLI Example)

These commands configure an additional set of real servers and a service group for duplicate Accounting-Request messages:

```
ACOS(config)# slb server diameter-acr-dup1 10.10.10.4
ACOS(config-real server)# port 3868 tcp
ACOS(config-real server-node port)# health-check-disable
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
ACOS(config)# slb server diameter-acr-dup2 10.10.10.5
ACOS(config-real server)# port 3868 tcp
ACOS(config-real server-node port)# health-check-disable
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
ACOS(config)# slb service-group diameter-duplicate-group tcp
```

```
ACOS(config-slb svc group)# method round-robin-strict  
ACOS(config-slb svc group)# member diameter-acr-dup1 3868  
ACOS(config-slb svc group-member:3868)# exit  
ACOS(config-slb svc group)# member diameter-acr-dup2 3868  
ACOS(config-slb svc group-member:3868)# exit  
ACOS(config-slb svc group)# exit
```

The following commands add the duplicate option to the Diameter template:

```
ACOS(config)# slb template diameter diameter1  
ACOS(config-diameter template)# duplicate 30 "acct" diameter-duplicate-  
group  
ACOS(config-diameter template)# exit
```

This option duplicates Accounting-Request messages with AVP 30 that contain the string "acct", and sends the duplicate messages to service group "diameter-duplicate-group".

The duplicate service group does not need to be bound to the Diameter virtual port. Instead, the **duplicate** option in the Diameter template takes care of this.

RADIUS Message Load Balancing

This section describes RADIUS message load balancing and how to configure it.

The following topics are covered:

Overview	750
Configuration	753

Overview

This section describes the RADIUS Message Load Balancing information.

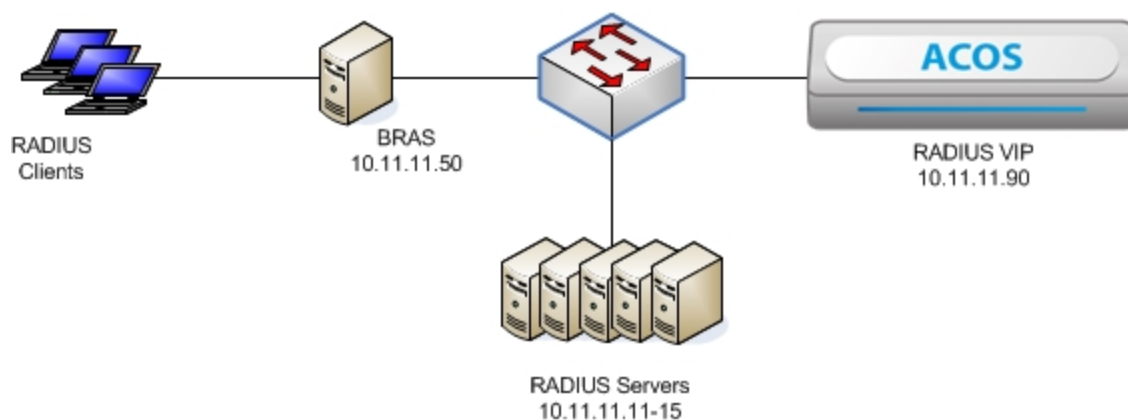
The following topics are covered:

Deployment	750
Server Traffic Flow	751
Stateless Load Balancing Protocol Ports Supported	752
RADIUS Session Aging	753

Deployment

ACOS supports load balancing of RADIUS messages in a topology such as the one shown in [Figure 75](#).

Figure 75 : RADIUS message load balancing



In this example, all RADIUS requests received by the ACOS device have the same source and destination IP addresses. The source address is the address of a Broadband Remote Access Server (BRAS), 10.11.11.50. The destination IP address is a RADIUS VIP on the ACOS device, 10.11.11.90.

In this type of topology, wherein RADIUS requests for multiple clients arrive at the ACOS device with the same source and destination addresses, using a UDP virtual

port does not provide load balancing. This is because the ACOS device uses the same session for all the requests.

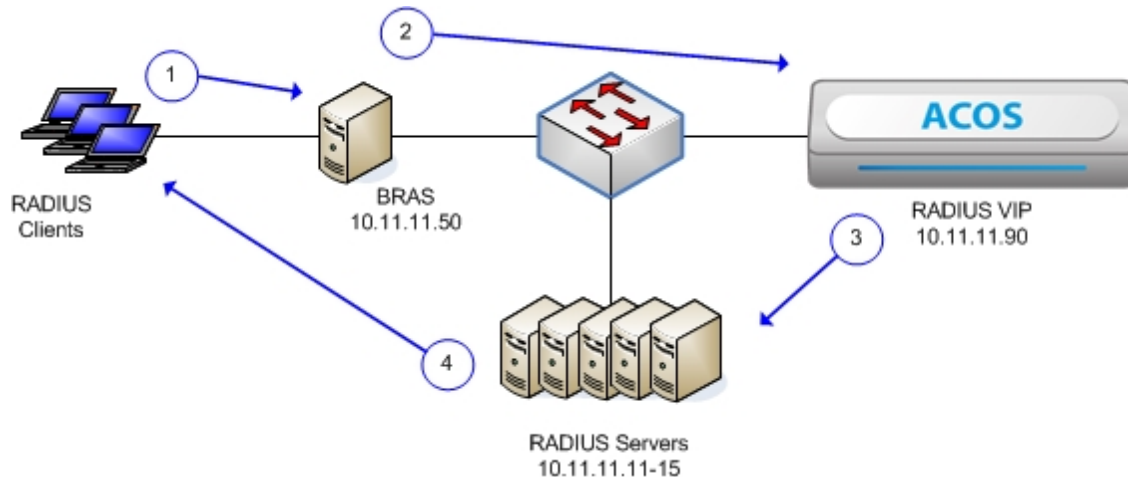
Normally, the ACOS device sends all traffic on a given session to the same server. If a UDP virtual port is used, the ACOS device uses the configured load balancing method to select a server for the first request, then uses the same server (and data CPU) for all subsequent requests.

Server Traffic Flow

The first time the ACOS device receives a RADIUS request with a given Identifier value, the device uses the load balancing method to select a RADIUS server. Subsequent RADIUS requests with the same Identifier go to the same server. However, when the ACOS device receives a request with a different Identifier value, the device uses the configured load balancing method to select a server for requests containing that value.

[Figure 76](#) shows the traffic flow for RADIUS message load balancing.

Figure 76 : RADIUS message load balancing - traffic flow



1. Client sends RADIUS request.
2. Request is forwarded by BRAS.
3. RADIUS VIP on ACOS device receives the request. ACOS device selects a server and sends the request to the server. Subsequent requests with the same

Identifier go to the same server.

4. RADIUS server replies to request.

Stateless Load Balancing Protocol Ports Supported

Both the virtual port number and real port numbers must be in the list to support stateless load balancing. Stateless load balancing for RADIUS is supported for the following protocol port numbers and ranges:

- 1645
- 1646
- 1812
- 1813
- 12,000 – 28,000
- 42,000 – 58,000

NOTE:

- Stateless load balancing for RADIUS on non-FPGA models can use any port numbers.
- For Hardware-Based Load-Balancing used on the FPGA model, configure the port number as per the following port ranges:
 - Virtual port number within 42,000 – 58,000 or 1645/1646/1812/1813
 - Real port number within 12000 – 28000 or 1645/1646/1812/1813

For details, refer to [Hardware-Based Load-Balancing Methods](#).

- On ACOS models that use FTAs, using stateful and stateless load-balancing methods simultaneously is not supported for the listed protocol ports if the same port number is used for real and virtual ports.
-

RADIUS Session Aging

Aging of RADIUS sessions is based on the idle-timeout in the UDP template used by the RADIUS virtual port. The default is 120 seconds.

You also can use the aging option in the UDP template. For example, setting aging to “immediate” terminates a session as soon as the server responds to the client.

Configuration

Configuration of RADIUS message load balancing is similar to configuration of other service types.

Configuration RADIUS Message Load Balancing (Procedure)

1. (Optional) To configure connection-rate limiting or request-rate limiting for the RADIUS ports, configure a real-port template and set the rate within the template.
2. Create a real server configuration for each RADIUS server.
 - Make sure the port number(s) you assign to the RADIUS port(s) are among those listed in [Stateless Load Balancing Protocol Ports Supported](#).
 - If you configure a real-port template (step 1 above), bind the template to each of the RADIUS ports in each real-server configuration.
3. Add the real servers to a service group. Configure the group to use the Stateless Per-packet Round-robin load-balancing method.
4. Create a VIP configuration that has the IP address that will receive the RADIUS requests. Add a RADIUS virtual port to the VIP. Bind the service group created in step 3 to the virtual port.
5. The RADIUS port number on each real server must be the same. Using mixed port numbers in a service group is not supported. The virtual port number does not need to be the same as the real port number.

Configuration RADIUS Message Load Balancing (CLI Procedure)

At the configuration level for the virtual server, use the **port** command. To view RADIUS sessions, use the **show session radius** command.

Configuration RADIUS Message Load Balancing (CLI Example)

The commands in this example implement the RADIUS load balancing configuration shown in [RADIUS message load balancing](#) and [RADIUS message load balancing - traffic flow](#).

To begin, the following commands create server configurations for the RADIUS servers:

```
ACOS(config)# slb server radius1 10.11.11.11
ACOS(config-real server)# port 1812 udp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
ACOS(config)# slb server radius2 10.11.11.12
ACOS(config-real server)# port 1812 udp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
ACOS(config)# slb server radius3 10.11.11.13
ACOS(config-real server)# port 1812 udp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
ACOS(config)# slb server radius4 10.11.11.14
ACOS(config-real server)# port 1812 udp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
ACOS(config)# slb server radius5 10.11.11.15
ACOS(config-real server)# port 1812 udp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
```

The following commands add the servers to a service group:

```
ACOS(config)# slb service-group sg-rad udp
ACOS(config-slb svc group)# member radius1 1812
ACOS(config-slb svc group-member:1812)# exit
ACOS(config-slb svc group)# member radius2 1812
ACOS(config-slb svc group-member:1812)# exit
```

```

ACOS(config-slb svc group)# member radius3 1812
ACOS(config-slb svc group-member:1812)# exit
ACOS(config-slb svc group)# member radius4 1812
ACOS(config-slb svc group-member:1812)# exit
ACOS(config-slb svc group)# member radius5 1812
ACOS(config-slb svc group-member:1812)# exit
ACOS(config-slb svc group)# exit

```

The following commands configure the RADIUS VIP:

```

ACOS(config)# slb virtual-server rad-msg-lb 10.11.11.90
ACOS(config-slb vserver)# port 1812 radius
ACOS(config-slb vserver-vport)# service-group sg-rad
ACOS(config-slb vserver-vport)# exit
ACOS(config-slb vserver)# exit

```

The following command shows active RADIUS sessions:

```
ACOS# show session radius
```

Traffic Type	Total
-----	-----
TCP Established	0
TCP Half Open	0
UDP	30
...	

Prot	Forward Source	Forward Dest	Reverse Source
Reverse Dest	Age	Hash Flags Radius ID	
-----	-----	-----	-----
Udp	10.11.11.50:32836	10.11.11.90:1812	10.11.11.15:1812
10.11.11.50:32836	120	1 NSe0 104	
Udp	10.11.11.50:32836	10.11.11.90:1812	10.11.11.12:1812
10.11.11.50:32836	120	1 NSe0 111	
Udp	10.11.11.50:32836	10.11.11.90:1812	10.11.11.15:1812
10.11.11.50:32836	120	1 NSe0 167	
Udp	10.11.11.50:32836	10.11.11.90:1812	10.11.11.14:1812
10.11.11.50:32836	120	1 NSe0 118	
Udp	10.11.11.50:32836	10.11.11.90:1812	10.11.11.11:1812
10.11.11.50:32836	120	2 NSe0 70	
Udp	10.11.11.50:32836	10.11.11.90:1812	10.11.11.14:1812
10.11.11.50:32836	120	2 NSe0 91	

Udp	10.11.11.50:32836		10.11.11.90:1812	10.11.11.15:1812
	10.11.11.50:32836	120	2 NSe0 84	
Udp	10.11.11.50:32836		10.11.11.90:1812	10.11.11.14:1812
	10.11.11.50:32836	120	2 NSe0 77	
Udp	10.11.11.50:32836		10.11.11.90:1812	10.11.11.15:1812
	10.11.11.50:32836	120	2 NSe0 56	
Udp	10.11.11.50:32836		10.11.11.90:1812	10.11.11.12:1812
	10.11.11.50:32836	120	2 NSe0 119	
Udp	10.11.11.50:32836		10.11.11.90:1812	10.11.11.13:1812
	10.11.11.50:32836	120	2 NSe0 168	
Udp	10.11.11.50:32836		10.11.11.90:1812	10.11.11.13:1812
	10.11.11.50:32836	120	3 NSe0 162	
Udp	10.11.11.50:32836		10.11.11.90:1812	10.11.11.13:1812
	10.11.11.50:32836	120	3 NSe0 176	
Udp	10.11.11.50:32836		10.11.11.90:1812	10.11.11.12:1812
	10.11.11.50:32836	120	3 NSe0 239	
Udp	10.11.11.50:32836		10.11.11.90:1812	10.11.11.13:1812
	10.11.11.50:32836	120	3 NSe0 15	
Udp	10.11.11.50:32836		10.11.11.90:1812	10.11.11.11:1812
	10.11.11.50:32836	120	3 NSe0 134	
Udp	10.11.11.50:32836		10.11.11.90:1812	10.11.11.11:1812
	10.11.11.50:32836	120	3 NSe0 169	
Udp	10.11.11.50:32836		10.11.11.90:1812	10.11.11.13:1812
	10.11.11.50:32836	120	3 NSe0 204	
Udp	10.11.11.50:32836		10.11.11.90:1812	10.11.11.11:1812
	10.11.11.50:32836	120	4 NSe0 233	
Udp	10.11.11.50:32836		10.11.11.90:1812	10.11.11.15:1812
	10.11.11.50:32836	120	4 NSe0 107	
Udp	10.11.11.50:32836		10.11.11.90:1812	10.11.11.12:1812
	10.11.11.50:32836	120	5 NSe0 171	
Udp	10.11.11.50:32836		10.11.11.90:1812	10.11.11.14:1812
	10.11.11.50:32836	120	5 NSe0 66	
Udp	10.11.11.50:32836		10.11.11.90:1812	10.11.11.11:1812
	10.11.11.50:32836	120	5 NSe0 199	
Udp	10.11.11.50:32836		10.11.11.90:1812	10.11.11.12:1812
	10.11.11.50:32836	120	6 NSe0 221	
Udp	10.11.11.50:32836		10.11.11.90:1812	10.11.11.15:1812
	10.11.11.50:32836	120	6 NSe0 172	


```

Udp    10.11.11.50:32836      10.11.11.90:1812      10.11.11.13:1812
10.11.11.50:32836      120    6    NSe0 151
Udp    10.11.11.50:32836      10.11.11.90:1812      10.11.11.14:1812
10.11.11.50:32836      120    6    NSe0 25
Udp    10.11.11.50:32836      10.11.11.90:1812      10.11.11.12:1812
10.11.11.50:32836      120    6    NSe0 179
Udp    10.11.11.50:32836      10.11.11.90:1812      10.11.11.14:1812
10.11.11.50:32836      120    7    NSe0 103
Udp    10.11.11.50:32836      10.11.11.90:1812      10.11.11.11:1812
10.11.11.50:32836      120    7    NSe0 222
Total Sessions:          30

```

The session table contains a separate session for each RADIUS Identifier value. The following address information is shown for each session:

- Forward Source – The sender of the RADIUS message. In [RADIUS message load balancing](#), this is the IP address of the BRAS.
- Forward Dest – The RADIUS VIP on the ACOS device.
- Reverse Source – The RADIUS server to which the ACOS device sends requests that have the Identifier listed in the RADIUS ID field.
- Reverse Dest – The destination of the RADIUS server reply forwarded by the ACOS device. (This is the sender of the initial RADIUS message that started the session, the BRAS in the example above.)

SNMP-Based Load Balancing

This section describes SNMP-based load balancing.

The following topics are covered:

Overview	759
Configuration	762

Overview

You can use the results of SNMP queries to real servers to dynamically set the weight values of the servers. When used along with a load-balancing method that includes server weight in the selection process, this option allows servers to be selected based on metrics such as the following:

- CPU utilization
- System memory utilization
- Connection capacity
- Hard drive utilization

Requirements

- External health monitor – SNMP-based load balancing uses an external health monitor (a script) to periodically send SNMP queries to each of the real servers. The script must return a numeric value. The following values are valid for SNMP-based load balancing:
 - 0-124 – Server is up. The server's weight value (1-100) is dynamically changed to the value returned by the script. If the script returns 0, the value is set to 1. If the script returns value 101-124, the value is set to 100.
 - -1 – Server is down.
 - 125 or higher – Server is down.
- Exit codes 1 and 2 are reserved. Please make sure the script does not have general errors.

When configuring the external health monitor on the ACOS device, make sure to use the preference option. This option enables the server weight values to be dynamically set based on the values returned by the health monitor's script.

- Weighted load-balancing method – The server weight is used for server selection only if the service group uses either of the following load-balancing methods:

- Weighted-least-connection
- Weighted-rr (weighted round robin)

Weight-based Load Balancing

When a weight-based load-balancing method is used, the ACOS device selects servers with higher weight values more often than servers with lower weight values.

For example, assume the SNMP-based health check of a group of 4 real servers results in the following dynamically assigned weight values:

- Server1 – weight 1
- Server2 – weight 2
- Server3 – weight 4
- Server4 – weight 5

ACOS selects the servers for new connections as follows:

- Server1 – 1 connection
- Server2 – 1 connection
- Server3 – 1 connection
- Server4 – 1 connection
- Server1 – 0 connections
- Server2 – 1 connection
- Server3 – 1 connection
- Server4 – 1 connection
- Server1 – 0 connections
- Server2 – 0 connections
- Server3 – 1 connection
- Server4 – 1 connection
- Server1 – 0 connections
- Server2 – 0 connections

- Server3 – 1 connection
- Server4 – 1 connection
- Server1 – 0 connections
- Server2 – 0 connections
- Server3 – 0 connections
- Server4 – 1 connection
- Server1 – 1 connection
- Server2 – 1 connection
- Server3 – 1 connection
- Server4 – 1 connection

...

The pattern of selection repeats until the server weight values are changed based on the next health check.

Health Monitoring

Here is an example of a Shell script for SNMP-based load balancing health monitoring. This script uses the results from queries to an example MIB module from ExampleCorp:

```
#!/bin/bash

dst="$HM_SRV_IPADDR"
#dst="$HM_SRV_IPADDR:$HM_SRV_PORT"

# Community String
community="public"

# EXAMPLECORP-SNMP-MIB::extResult.1
oid=".1.3.6.1.4.1.2021.8.1.100.1"

function check_value {
    echo "$output"
    value=`echo $output | sed 's/INTEGER: //'`
```

```
echo "value" = "$value"
if [[ "$value" =~ ^[[:digit:]]*$ ]]; then
#   echo "digit string"
   if [ $value -ge 0 -a $value -lt 125 ]; then
       echo "Good"
       exit $value
   fi
fi
}

echo "/usr/bin/snmpget -v2c -c \"\$community\" -Ov \"\$dst\" $oid"
output=$(/usr/bin/snmpget -v2c -c "$community" -Ov "$dst" $oid)
if [ 0 != $? ]; then
    exit -1
fi
check_value

# success
echo "Mark server down"
exit -1
```

Configuration

To configure SNMP-based load balancing:

1. Create a script such as the one shown above.
2. Import the script onto the ACOS device.
3. Configure an external health monitor to use the script. Make sure to use the **preference** option.
4. Configure the health check on the real server.
5. Configure the service group to use a weighted load-balancing method.

NOTE: These steps apply specifically to SNMP-based load balancing. The other configuration steps standard to all types of load balancing are also required: configure the real servers and add them to a service group, configure the virtual server (VIP), bind the service group to a virtual port on the VIP, and so on.

GUI Configuration

The section provides an example of how to configure SNMP-based load balancing using the GUI.

1. Import the script into the GUI:
 - a. Navigate to **ADC >> Health Monitors >> External Programs**.
 - b. Click **Create**.
 - c. Provide a name, then copy and paste the external script into the GUI.
 - d. Click **Create**.
2. Create the health monitor that will use the script.
 - a. Navigate to **ADC >> Health Monitors**.
 - b. Click **Create**.
 - c. In the General Fields section, specify a name, then select External from the drop-down list in the Method type field.
 - d. In the External section, select the name of the script you just created from the drop-down list in the Program field.
 - e. Select the checkbox in the Preference field.
 - f. Click **Create Monitor**.
3. Configure the service group to use a weighted load-balancing method.
 - a. Navigate to **ADC >> SLB >> Service Groups**.
 - b. Specify a service group name and protocol.
 - c. In the Algorithm field, select a weight load-balancing method (for example, Weighted Round Robin).

- d. Complete any other fields are needed (refer to the GUI online help for more information about any field on this screen).
- e. Click **Create**.

CLI Configuration

This section provides an example of SNMP-based load balancing using the CLI.

1. Import an external health monitor script onto the ACOS device using the **import health-external** command at the global configuration level. For example, to import a script named `snmp-hm.sh`:

```
ACOS(config)# import health-external snmp-hm.sh
scp://exampleuser@192.168.10.14/scripts/
```

2. Configure an external health monitor that will use the script. For example:

```
ACOS(config)# health monitor hm-ext-snmp
```

This command changes the CLI to the configuration level for the monitor, where you use the **method** command to have the health monitor use the external script:

```
ACOS(config-health:monitor)# method external program snmp-hm.sh
preference
```

3. Set the service group to use a load-balancing method based on server weight:

```
ACOS(config)# slb service-group snmp-sg1 tcp
ACOS(config-slb svc group)# method weighted-rr
```

4. To verify that all servers are up, use the **show health stat** command.
5. To display current weight values of the servers, use the **show slb server config** command.

Advanced Traffic Replication

The following topics are covered:

Overview	766
Configuration	773

Overview

The following topics are covered:

Topology	766
Dataflow	767
Packet Header Changes	769
Modes	770
Use Case Scenarios	771
Implementation Details	773

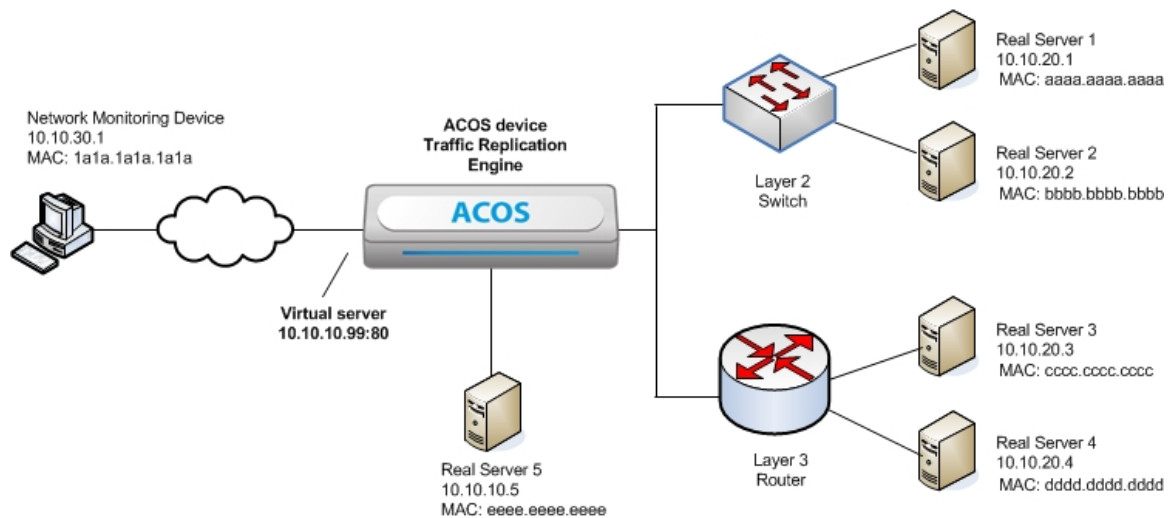
Topology

ACOS includes a traffic replication feature that intercepts traffic feeds, such as SNMP or Syslog packets, copies them to a buffer, and forwards the duplicated packet to multiple collector servers, where the data can be used to track users and devices.

The new feature can be helpful for organizations that need Network Monitoring feeds to be replicated to multiple destinations. It represents a significant improvement over alternative method that rely on servers processing feeds and then forwarding them to other groups in an organization.

[Figure 77](#) shows the topology used to discuss this traffic replication feature.

Figure 77 : Topology for Traffic Replication

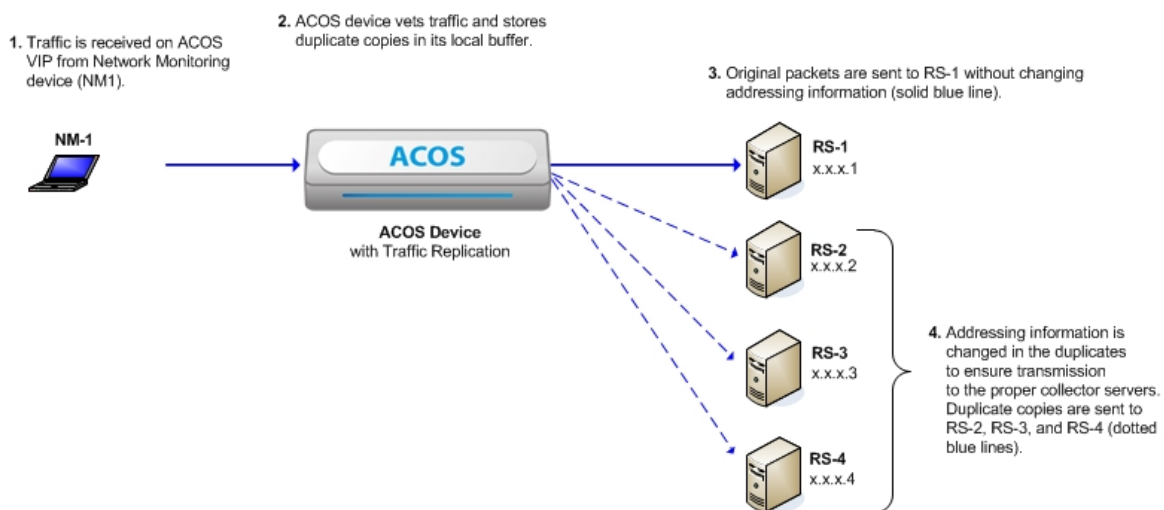


The figure shows an ACOS device connected to multiple real “collector” servers. The servers can be connected directly to the ACOS device (Server 5), or they can be connected through a Layer 2 switch (Servers 1 and 2), or they can be connected through a Layer 3 router (Servers 3 and 4).

Dataflow

[Figure 78](#) shows how the traffic replication feature works.

Figure 78 : Traffic Replication Data Flow



1. The Network Monitoring device (NM-1) sends traffic to the ACOS VIP.
2. As traffic passes through the ACOS device, it is vetted to see if it belongs to one of the target NM traffic types, such as SNMP or Syslog. If the traffic *does* belong to one of the NM traffic types, then duplicates are made for each collector server and are stored locally on the ACOS device.
3. The original traffic from NM-1 is load balanced using standard SLB algorithms and is sent to its original target destination (RS-1). This is represented by the solid blue line in [Figure 78](#).
4. ACOS applies one traffic replication option to the duplicated packets. This customization of duplicated packet changes the MAC or IP in the packet's header. These changes forward the packets to multiple destinations (RS-2, RS-3, and RS-4). Forwarded packets are represented by the dotted blue lines.

While previous releases supported a port mirroring feature, it operated at the port level and did not discriminate between different types of traffic. This new approach to traffic replication provides better granularity by enabling you to specify which parts of the source and destination addresses will be replaced.

The feature allows you to bind a traffic replication mode to a normal VIP or to a wildcard VIP, and that wildcard VIP can be associated with an ACL.

Separate VIPs can be created on the ACOS device to handle specific types of traffic. For example, a VIP could be dedicated to receiving SNMP traffic. When traffic is received on that VIP, (and assuming one of the replication modes has been enabled), the SNMP traffic stream will be replicated to the collector servers designated within the associated service group.

Traffic replication supports both UDP and TCP protocols. However, TCP replication introduces an important consideration.

For TCP replication to function correctly, the replication or collector servers must passively receive packets. ACOS does not expect a TCP SYN-ACK response from replication servers. If a collector server responds to a replicated TCP packet, for example a SYN-ACK, ACOS will issue a TCP RST, terminating the connection and causing replication failure.

To enable TCP-based replication effectively, a primary collector server must be assigned with a higher priority, which will handle the full TCP handshake. All other

collector servers must be lower priority and configured to passively receive replicated traffic only.

For example, in the following configuration, s1 completes the TCP handshake and actively receives the traffic. The servers s2 and s3 passively receive the duplicated packets. No TCP responses should be sent by s2 or s3, or replication will fail due to a TCP RST from ACOS:

```
slb service-group syslog_nms tcp
  traffic-replication-type mirror-ip-repl
    member s1:514 priority 10
    member s2:514 priority 1
    member s3:514 priority 1
```

Packet Header Changes

Most of traffic replication options modify headers of duplicated packets at Layer 2 by changing the MAC address. As [Figure 79](#) shows, traffic replication mainly affects the packet addressing at Layer 2 and Layer 3.

Figure 79 : Changes to Packet Header

Traffic Replication changes mainly affect L2 and L3, with very minor changes to L4 source and dest ports.

OSI Layer	Source	Destination
L4	Source Port	(No change to payload)
	Dest Port	
L3	Source IP	Dest IP
L2	Source MAC	Dest MAC
L1	(No changes)	(No changes)

Only one of the Traffic Replication modes, mirror IP-replacement, alters the packets' IP address and makes changes to the Layer 4 source and destination ports in the duplicated packets.

- When using the mirror IP-replacement option, the *destination* port can be changed under the following circumstances:
 - If the virtual port is set to wildcard port 0, and if the service group members have different real ports configured, then the Layer 4 destination port on the duplicated packets will be changed.
 - If the virtual port is set to wildcard port 0, and if a service group member is configured with port 80, then the Layer 4 destination port on the duplicated packets will be changed to port 80.
 - If the virtual port is set to wildcard port 0, and if a service group member is configured with real port 0, then the Layer 4 destination port will *not* be changed.
 - If the virtual port is *not* set to wildcard port 0 but is instead set to port 80, and if a service group member is configured with real port 81, then the Layer 4 destination port will be changed to port 81.
- When using the mirror IP-replacement option, the *source* port can be changed under the following circumstances:
 - The Layer 4 source port will only be changed if the original packet being load balanced and replicated is changed. The reasons for this change to the source port of the original packet, (and in the duplicated packets) are unrelated to the Advanced Traffic Replication feature.
 - Source NAT can be used with the mirror IP-replacement option. In this case, the Layer 4 source-port might be replaced for packets that have been load balanced, but all of the replicated packets will have the same source port as the packet that was load balanced.

Modes

Traffic can be replicated or “mirrored” to the collector servers that are specified within a service group using one of the following traffic replication modes:

- Mirror: This mode does not change the packet header at all. The original Layer 2 Destination Address (DA) or Source Address (SA) and Layer 3 IP addresses are left intact. ACOS simply sends the packets “as is” to the collector server(s), and the forwarding is based on the IP address in the original packet.

Unlike other replication modes, mirror mode replicates traffic to the client, in addition to replicating traffic to the server. Only lower priority servers are used, so it is recommended to define the collector servers as lower priority to ensure they receive the replicated traffic.

For example, when SLB configuration uses the option `traffic-replication-type mirror-ip-repl`, the ACOS device should load balance traffic to the higher-priority service-group members and replicate traffic to the lower priority members or collectors.

- **Mirror Destination MAC Address replacement:** This mode uses Layer 2 forwarding, with the ACOS device replacing the destination MAC address on the incoming packet with the destination MAC for each of the collector servers within the designated service group.
- **Mirror IP-replacement:** This mode replaces the incoming packet's IP address with the IP address of the collector server(s) and forwards the duplicated packet to those servers. This is the only mirroring option that affects the packet at Layer 4, with minor changes being made to the Layer 4 source and destination ports. This option is recommended for scenarios in which collector servers are not directly connected to the ACOS device.
- **Mirror Source MAC Address and Destination MAC Address replacement:** This mode replaces both source and destination MAC addresses (Layer 2); does *not* change IP addressing information (Layer 3).
- **Mirror Source MAC Address replacement:** This mode replaces the source MAC address on the incoming packet with the MAC address corresponding to virtual server on the ACOS device. This option is recommended for scenarios where not changing the source MAC can cause a loop.

Use Case Scenarios

[Table 16](#) lists replication modes and associated use case scenarios for each mode.

Table 16 : Traffic Replication Options and Use Cases

Traffic Replication Option	CLI Description	Use Case Scenario
mirror	Mirror Bi-directional	If UDP Network Management traffic

Table 16 : Traffic Replication Options and Use Cases

Traffic Replication Option	CLI Description	Use Case Scenario
	Packet	collectors operate in promiscuous mode, then this mirror option can be used to replicate traffic to collectors that are directly connected to ACOS.
mirror-da-repl	Replace Destination MAC	When the collector server is connected through a Layer 2 network, or if the collector server is not operating in promiscuous mode, this option can be used to require the destination MAC to be set with the collector server's MAC.
mirror-ip-repl	Replaces IP with server-IP	If UDP Network Management traffic collectors cannot operate in promiscuous mode, this “mirror-ip-rep” option can be used to replicate traffic to collectors that are not directly connected to the ACOS device and which are on a different subnet. This can be particularly useful with event management applications (such as netcool), which employ an active/standby architecture and replicate traffic between the servers. This application can be offloaded from the servers and onto the ACOS device, thus improving performance.
mirror-sa-da-repl	Replace Source MAC and Destination MAC	This option is similar to the “mirror-da-repl” option because the destination MAC is replaced. In addition to the destination MAC being replaced, the source MAC will also be changed to the ACOS device's MAC address to prevent network loops.

Table 16 : Traffic Replication Options and Use Cases

Traffic Replication Option	CLI Description	Use Case Scenario
mirror-sa-repl	Replace Source MAC	This option is similar to the "mirror" option because the source MAC address is changed in order to provide additional protection and for identification purposes.

Implementation Details

Implementing the Traffic Replication feature is almost the same set of configuration steps as required for a standard SLB. To configure this feature, the following configurations are necessary:

1. Define a normal VIP (or a wildcard VIP) within the service group. If a wildcard VIP is used, then create an ACL to identify the subnet of the network monitoring devices from which traffic will be accepted.
2. Configure the real collector servers.
3. Configure a service group for the collector servers, add the real collector servers to the service group, and define which traffic replication modes are used with the `traffic-replication-type` command.

Configuration

Configuring Advanced Traffic Replication (CLI Procedure)

To configure a traffic replication mode, use the `traffic-replication-type` command (service-group level).

Configuring Advanced Traffic Replication (CLI Example)

The following commands define a VIP on the ACOS device.

```
ACOS(config)# slb virtual-server NM-VIP 10.10.10.99
```

```
ACOS(config-slb vserver)# exit
```

The following commands configure the real collector servers.

```
ACOS(config)# slb server RS-1 10.10.20.1  
ACOS(config-real server)# exit  
ACOS(config)# slb server RS-2 10.10.20.2  
ACOS(config-real server)# exit  
ACOS(config)# slb server RS-3 10.10.20.3  
ACOS(config-real server)# exit  
ACOS(config)# slb server RS-4 10.10.20.4  
ACOS(config-real server)# exit  
ACOS(config)# slb server RS-5 10.10.10.5  
ACOS(config-real server)# exit
```

These commands configure a service group for collector servers and add real collector servers to the service group. The `traffic-replication-type` command then specifies traffic replication mode used to forward duplicated network monitoring traffic to the collector servers.

```
ACOS(config)# slb service-group SG-RS tcp  
ACOS(config-slb svc group)# member RS-1 0  
ACOS(config-slb svc group-member:0)# priority 10  
ACOS(config-slb svc group-member:0)# exit  
ACOS(config-slb svc group)# member RS-2 0  
ACOS(config-slb svc group-member:0)# priority 10  
ACOS(config-slb svc group-member:0)# exit  
ACOS(config-slb svc group)# member RS-3 0  
ACOS(config-slb svc group-member:0)# priority 8  
ACOS(config-slb svc group-member:0)# exit  
ACOS(config-slb svc group)# member RS-4 0  
ACOS(config-slb svc group-member:0)# priority 5  
ACOS(config-slb svc group-member:0)# exit  
ACOS(config-slb svc group)# member RS-5 0  
ACOS(config-slb svc group-member:0)# priority 5  
ACOS(config-slb svc group-member:0)# exit  
ACOS(config-slb svc group)# traffic-replication-type mirror-da-repl  
ACOS(config-real server)# exit
```

Outbound Next Hop Load Distributor

The following topics are covered:

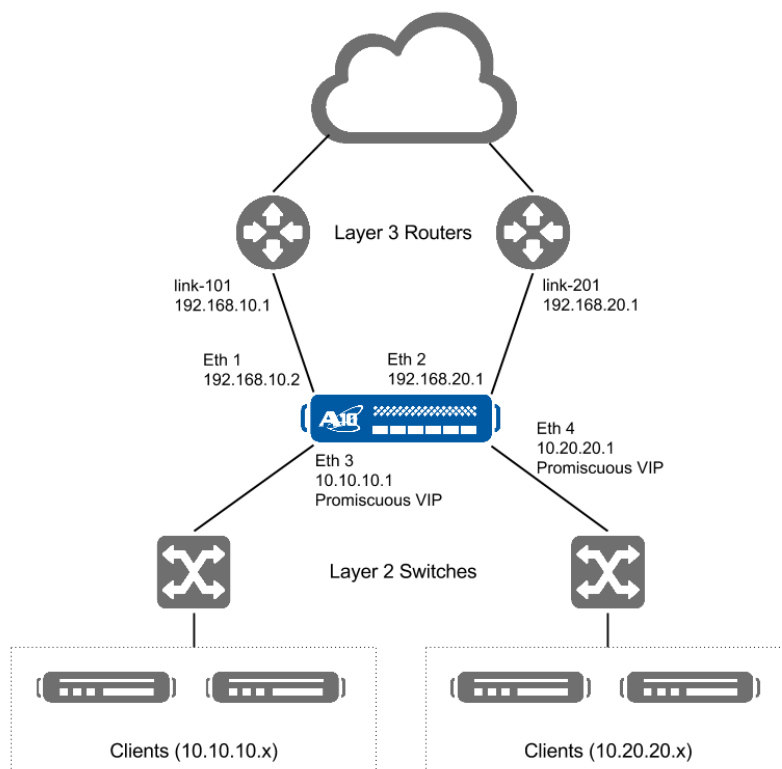
Next Hop Load Distributor Overview	776
Configuration	778

Next Hop Load Distributor Overview

ACOS supports outbound Next Hop Load Distributor (NHLD). Outbound NHLD enables you to balance client-server traffic across a set of WAN links. In outbound NHLD, the clients are located on the internal side of the network. The servers are located on the external side of the network.

[Figure 80](#) shows an example of outbound NHLD.

Figure 80 : Next Hop Load Distributor



In this example, the ACOS device is configured to balance client traffic across a set of two WAN links, through next-hop routers 192.168.10.1 and 192.168.20.1.

When the ACOS device receives a request from a client, the ACOS device uses SLB load balancing to select one of the WAN links. ACOS then uses source IP NAT to translate the client's private IP address into a public IP address, then sends the client's request to the next-hop router for the selected WAN link.

When the ACOS device receives the server's reply to a client request, the ACOS device translates the destination IP address from the NAT address to the client's private IP address, then forwards the reply to the client.

Load Balancing Methods

Use one of the following load balancing methods to load balance traffic across the WAN links:

- Round-robin – Selects link in simple rotation. This results in each link being selected an equal number of times. The default method is round-robin.
- Least-connections – Selects the link that has the least current client connections on it. The connection count is based on client connections initiated on the link by the ACOS device.
- Link-Cost – In this method, link usage is prioritized based on the cost of the bandwidth for the connection. If the link utilization exceeds the pre-configured bandwidth, the excess traffic is sent to the link operating at the lowest overage cost. Additionally, in case of a link failure, traffic is sent to the next available link.

Requirements

In an outbound NHLD topology, the next-hop routers for the WAN links must be able to send the server reply traffic back to the ACOS device. To ensure that the server reply traffic passes back through the ACOS device, use an IP source NAT pool for each WAN link.

The pools do not need to contain more than a few addresses. ACOS internally uses a separate protocol port number for each client session on a pool address.

SLB destination NAT, which is enabled by default, must be disabled. In a standard SLB configuration, destination NAT is used to translate the server address (destination IP address) requested by clients from the VIP address into the server's real address. However, this NAT operation is not applicable to outbound NHLD.

Configuration

Configuring Next Hop Load Distributor (Procedure)

1. Configure an IP source NAT pool for each link to be load balanced. The address range in a pool must be in the same subnet as the next-hop router's interface with the ACOS device.
2. Configure a pool group and add the pools to it.
3. Configure ACOS interfaces connected to the clients. Enable promiscuous VIP support on the interfaces.
4. Configure the ACOS interfaces connected to the next-hop routers for the links to be load balanced. (Do not enable promiscuous VIP on these interfaces.)
5. Configure a real server for each load balanced link. Add wildcard ports (TCP, UDP, or both) to the servers.
 - Use Layer 3 health checking (ICMP ping) to check router's IP interface health. When testing a path *through* a device between the ACOS device and router, use the ICMP health monitor transparent option.
 - The configuration requires health checking to be *disabled* on the wildcard ports added for a router. The router will not respond to these health checks. If you leave health checking enabled on the wildcard ports, the ACOS device will mark the ports down and NHLD will not work.
6. Configure a service group for the links (real servers). If the real server configurations for the links have both TCP and UDP ports, configure a service group for TCP and another service group for UDP.
7. Configure a virtual server with virtual IP address 0.0.0.0 (the wildcard VIP address). Using the wildcard VIP address enables the configuration to work for any destination IP address requested by clients.
 - Add the wildcard TCP port (TCP 0) and bind it to the TCP service group. Likewise, add the wildcard UDP port and bind it to the UDP service group.
 - Bind the ports to service group(s). On each port, bind the port to the IP Source NAT pool group and disable destination NAT.

Configuring Next Hop Load Distributor (CLI Example)

Commands in this example implement the NHLD configuration shown in [Next Hop Load Distributor](#). This example uses a single Ethernet port for each interface to the clients and the next-hop routers. You also can use trunk interfaces, virtual Ethernet (VE) interfaces, or both.

1. The following commands configure the IP source NAT pools and pool group:

```
ACOS(config)# ip nat pool nat10 192.168.10.3 192.168.10.4 netmask /24
ACOS(config)# ip nat pool nat20 192.168.20.3 192.168.20.4 netmask /24
ACOS(config)# ip nat pool-group outbound-nat-group
ACOS(config-pool-group:outbound-nat-gro)# member nat10
ACOS(config-pool-group:outbound-nat-gro)# member nat20
ACOS(config-pool-group:outbound-nat-gro)# exit
```

2. The following commands enable promiscuous VIP support on the ACOS interfaces connected to clients.

```
ACOS(config)# interface ethernet 3
ACOS(config-if:ethernet:3)# ip address 10.10.10.1 255.255.255.0
ACOS(config-if:ethernet:3)# ip allow-promiscuous-vip
ACOS(config-if:ethernet:3)# exit
ACOS(config)# interface ethernet 4
ACOS(config-if:ethernet:4)# ip address 10.20.20.1 255.255.255.0
ACOS(config-if:ethernet:4)# ip allow-promiscuous-vip
ACOS(config-if:ethernet:4)# exit
```

3. These commands configure the ACOS interfaces to the next-hop routers for the load-balanced links:

```
ACOS(config)# interface ethernet 1
ACOS(config-if:ethernet:1)# ip address 192.168.10.2 255.255.255.0
ACOS(config-if:ethernet:1)# exit
ACOS(config)# interface ethernet 2
ACOS(config-if:ethernet:2)# ip address 192.168.20.2 255.255.255.0
ACOS(config-if:ethernet:2)# exit
```

4. The following commands configure a real server for each link to be load balanced:

```
ACOS(config)# slb server link-101 192.168.10.1
ACOS(config-real server)# port 0 tcp
```

```
ACOS(config-real server-node port)# health-check-disable
ACOS(config-real server-node port)# exit
ACOS(config-real server)# port 0 udp
ACOS(config-real server-node port)# health-check-disable
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
ACOS(config)# slb server link-201 192.168.20.1
ACOS(config-real server)# port 0 tcp
ACOS(config-real server-node port)# health-check-disable
ACOS(config-real server-node port)# exit
ACOS(config-real server)# port 0 udp
ACOS(config-real server-node port)# health-check-disable
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
```

5. The following commands configure service groups for the links:

```
ACOS(config)# slb service-group outbound-tcp-links tcp
ACOS(config-slbg svc group)# member link-101 0
ACOS(config-slbg svc group-member:0)# exit
ACOS(config-slbg svc group)# member link-201 0
ACOS(config-slbg svc group-member:0)# exit
ACOS(config-slbg svc group)# exit
ACOS(config)# slb service-group outbound-udp-links udp
ACOS(config-slbg svc group)# member link-101 0
ACOS(config-slbg svc group-member:0)# exit
ACOS(config-slbg svc group)# member link-201 0
ACOS(config-slbg svc group-member:0)# exit
ACOS(config-slbg svc group)# exit
```

6. The following commands configure the virtual server:

```
ACOS(config)# slb virtual-server wildcard-vip 0.0.0.0
ACOS(config-slbg vserver)# port 0 tcp
ACOS(config-slbg vserver-vport)# service-group outbound-tcp-links
ACOS(config-slbg vserver-vport)# source-nat pool outbound-nat-group
ACOS(config-slbg vserver-vport)# no-dest-nat
ACOS(config-slbg vserver-vport)# exit
ACOS(config-slbg vserver)# port 0 udp
ACOS(config-slbg vserver-vport)# service-group outbound-udp-links
```



```
ACOS(config-slb vserver-vport) # source-nat pool outbound-nat-group  
ACOS(config-slb vserver-vport) # no-dest-nat  
ACOS(config-slb vserver-vport) # exit  
ACOS(config-slb vserver) # exit
```

Configuration Based on Latency

This feature defines a new load balancing method to select a link with the lowest latency among multiple links to the Internet.

The configuration consists of a wild card VIP which is bound to servers. The ACOS device receives traffic on the wild card VIP with destination IP as a remote Internet IP address. The ACOS device is required to select a link (configured as SLB server) with the least average RTT.

The RTT value is measured independently by probe traffic (HTTP GET requests) sent to a specific remote target IP through SLB servers. Each SLB server in the service-group configured using the command "**next-hop-link**". The SLB server (link) in this context is actually a router. Such probe traffic sent through each link (configured as SLB server) will have the destination IP as the remote target IP and the destination MAC of the SLB server.

The remote target IP is configured with the "**destination hostname** *<domain-name>* *<target-ip-options>*" command inside the **link-probe** template.

A customized URL can be configured using the "**url**" parameter in the link-probe template.

When target IP is resolved using the configured hostname instead of a static IP, then the user needs to configure a DNS server using the global configuration commands "**ip dns primary/secondary**" for resolution. The interval used for DNS queries is 60 seconds and the resolved IP for the hostname can be checked using "**show ip dns-cache**" and "**show slb link-probe entry** *<options>*" commands. If the DNS response contains multiple IP address, then the first IP address in the DNS response is used for sending the probe traffic.

Probe traffic sent and received from each SLB server will be recorded for round trip time.

By default, HTTP response with status code of 200 OK is considered for RTT calculation. User can also specify custom status codes or ranges of status codes to be considered for RTT calculation using the "`expected-status-code`" parameter in the link-probe template.

A maximum of ten samples can be considered per test. Number of samples can be changed

considering the "`probes-per-test`" parameter in the link-probe template. The interval at which

subsequent probes are sent out can be changed using the "`probe-interval`" parameter in the link-probe template. The default probe-interval is five seconds.

Once the configured number of probes per test are sent, a different interval named "`test-interval`" is used before the set of probes for the next test are sent out. The test-interval can be changed with the "`test-interval`" parameter in the link-probe template. The default test-interval is 60 seconds.

Each test calculates the average in RTT between the GET request and response from the destination.

In the current implementation, the probe traffic always is a HTTP connection regardless of the configured RTT calculation method.

RTT calculation can be done in two ways:

- Time between HTTP request and response (default).
- TCP Smoothed Round Trip Time (SRTT), that is calculated for the TCP connection up to the point of receiving an expected HTTP status code response.

The configuration is classified into two parts:

1. Mandatory configuration:

- link-probe template
- hostname

- destination IP (Static or DNS resolution) IPv4 and IPv6 supported
- bind the link-probe template to method "next-hop-link" under service-group

2. Optional configuration:

- test-interval
- probe-interval
- probes-per-test
- rtt-method
- URL (default is /)
- expected-status-code

Configure NHLD (CLI)

Configure Link-probe Template:

To configure a probe template use the following commands:

```
ACOS (config)# slb template link-probe xyz
ACOS (config-probe template)# destination hostname
Al(config-probe template)# destination hostname NAME-043 static-ipv6-addr
2001:203::5/64
```

Optionally, use one of the following commands to configure the probe template:

```
ACOS (config-probe template)# test-interval 4444
ACOS (config-probe template)# probe-interval 4444
ACOS (config-probe template)# probes-per-test 3
ACOS (config-probe template)# rtt-method http-rtt
ACOS (config-probe template)# rtt-method tcp-srtt
ACOS (config-probe template)# expected-status-code
```

Configure a Service Group:

1. Enter the **slb service-group** command at the global configuration level.
2. Enter the **member** command at the configuration level for the service group.
3. Enter the command **method next-hop-link link-probe-template < link-probe-template -name>**

```
ACOS (config)# slb service-group NAME-080 tcp
```

```
ACOS (config-slb svc group)# method next-hop-link link-probe-template  
NAME-060  
ACOS (config-slb svc group)# member rs211 0  
ACOS (config-slb svc group)# member rs211-3 0  
ACOS (config-slb svc group)# member rs212 0
```

HTTP Proxy

This section provides an overview of HTTP proxy, available configuration resources, and how to configure it.

The following topics are covered:

Overview	786
Configuring HTTP Proxy	789
Logging HTTP Proxy	842
Displaying HTTP Statistics	845
Proxy Chaining SLB	850
Explicit Proxy with AAM Policy	855
Explicit Proxy with Virtual Wire Interface	862
Explicit Proxy with Dual-Stack Forwarding Support	863
Explicit Proxy Without SSLi	870
Terminating SSLi Session Silently	871
Transparent Proxy with Normal VIPs Associated to ACL	873

Overview

A proxy is an agent that acts in place of the original requester. For a transparent proxy, the client is not aware of the use of a proxy (proxy server). In the case of an explicit proxy, client browsers are configured to send requests to a proxy server, hence the name explicit proxy as the proxy service is known.

When configuring a proxy, the main configuration difference between an explicit proxy and transparent proxy is that for an explicit proxy, the virtual ip address (VIP) is specified for the slb virtual-server template configured with an HTTP virtual port, whereas for a transparent proxy, the virtual server VIP is set as a wildcard (0.0.0.0) and must have a vport for HTTP and another for HTTPS traffic, and a separate policy-template for HTTP and HTTPS. Separate service-groups for HTTP and HTTPS are also needed to correctly route the two types of traffic.

Example of an explicit proxy virtual server template configuration:

```
slb virtual-server vip 192.0.2.2
  port 80 http
  service-group sg
  template policy https
  template dynamic-service ds
```

Example of a transparent proxy virtual server template configuration:

```
slb virtual-server wildcard 0.0.0.0
  port 80 http
    service-group HTTP
    template policy HTTP-policy
    template dynamic-service DNS
  port 443 https
    service-group SSL_8080
    template policy SSL-policy
    template dynamic-service DNS
    template client-ssl SSL
```

The main configuration element of an HTTP proxy is the policy template where the source rules and actions are defined in the forward-policy sub-configuration. The template is then bound to an `slb virtual-server` template with an HTTP virtual

port for explicit proxy and transparent proxy, and also an HTTPs virtual port for transparent proxy.

Explicit and transparent HTTP proxy services can be configured to operate separately, or in parallel. Proxy services also work in an SSLi configuration.

You can use the ACOS Device as an HTTP proxy to control client access to hosts based on lists of allowed traffic sources (clients) and destinations (Web servers). In the case of explicit proxies, client applications, which are typically Web browsers, must explicitly configure the proxy's IP address and port such that all HTTP requests will be sent to the explicit proxy.

When this feature is enabled, an HTTP virtual port on the ACOS Device intercepts HTTP requests from clients, validates the sources and the destinations, and forwards only those requests that come from valid sources and that are sent to approved destinations. Destinations are validated based on layer 3 IP addresses, HTTP URL/hostname strings or `web-category`. The policy template specifies the action to take to handle requests.

The destinations requested by clients can be filtered based on the URL of the request or the hostname in the Host header of the request.

- If both the source and destination are allowed, ACOS translates the client address into a NAT address, if applicable, and forwards the request to the destination.
- If either the source or the destination is not explicitly allowed by the applicable source or destination list, the request is dropped.

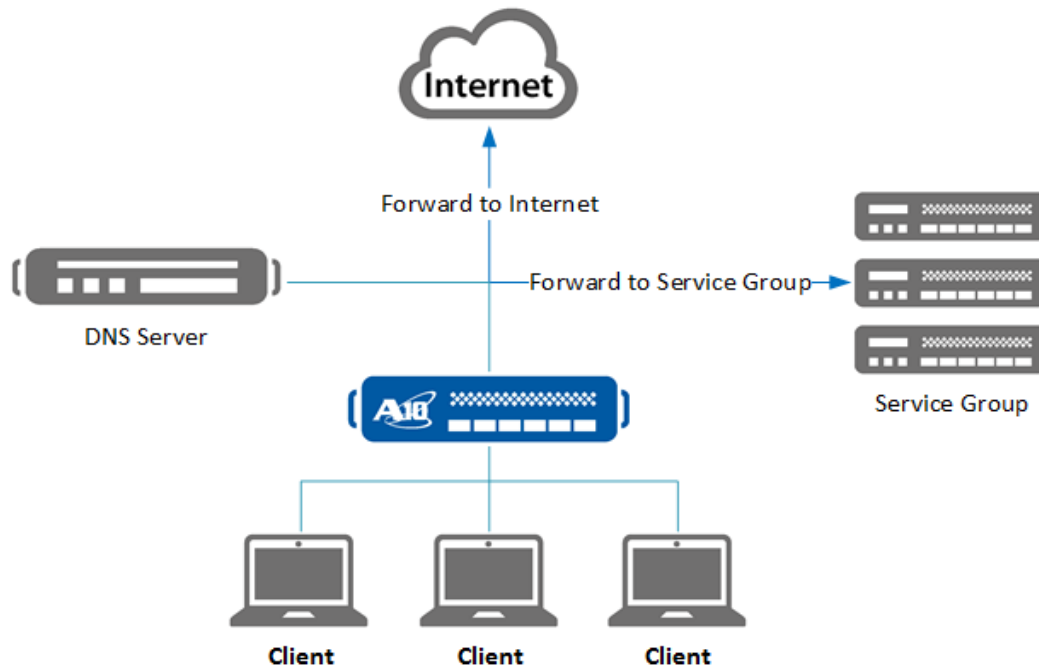
There are three mechanisms by which explicit HTTP proxy can forward traffic: traffic may be forwarded to the Internet, to a specified service group, or to another HTTP proxy server. These configuration scenarios are highlighted in this document.

The following differences between an explicit proxy vs. a transparent proxy are:

- An explicit proxy requires that clients are configured to point to the explicit proxy, whereas a transparent proxy requires no configuration in this regard.
- An explicit proxy offers more flexibility for managing traffic, although at a cost of additional configuration and resources, as you can create multiple VIPs, but can only have one wildcard VIP.

For a typical transparent proxy configuration, a vport is required for HTTP and another for HTTPs traffic, configured under an `slb virtual-server` template. This also means a separate policy and service-group are needed to handle each type of traffic.

Figure 81 : Explicit HTTP Proxy Mechanisms: Forward to Service Group and Forward to Internet



Explicit Proxy and Secure Sockets Layer Insight (SSLi) Integration

Explicit Proxy with Secure Sockets Layer Insight (SSLi) has been integrated to function together as one logical entity. Information on this can be found in the “Static-Port SSLi with Explicit and Transparent Proxy” in the SSL Configuration Guide.

Explicit Proxy Authentication Support

Application Access Management can be applied to an explicit proxy server. Information on this can be found in “Tracking Sessions” in the Application Access Management Guide.

Configuring HTTP Proxy

This section provides HTTP proxy-related configuration examples and resources.

The following topics are covered:

Resources Parameters	789
Basic Transparent Proxy Configuration	792
Basic Explicit Proxy Configuration	796
Advanced Explicit Proxy Configuration	805
Explicit Proxy URL Filtering	818
Transparent Proxy URL Filtering	822
Resolve Unknown URLs	837
Customizable Response Codes on Blocking HTTP Requests	839

Resources Parameters

To provide precise control, a forward policy defined within the SLB policy template is used to identify the sources, destinations, and actions for matching traffic. [Table 17](#) describes these resources and how they are related. For further clarification, see the configuration examples.

Table 17 : Configuration Resources for Explicit HTTP Proxy

Resource	Purpose	Description
Policy template	Applies actions to client-to-server traffic based on source and destination	Set of rules that define permitted traffic sources and destinations, and actions to apply to permitted traffic. This is defined within the SLB command. The explicit HTTP proxy relies only on the forward policy to define the source and destination match criteria, and corresponding actions to apply. For more information, see the forward policy description.
Forward	Specifies	Collection of source and destination match rules

Table 17 : Configuration Resources for Explicit HTTP Proxy

Resource	Purpose	Description
Policy	permitted traffic destinations and sources	based on source IP addresses, destination host/URLs of client requests, and actions to apply. Forward policy is a required component when configuring explicit HTTP proxy. Through this module, you can define the type and scope of the action desired for outbound traffic.
Source Match Rule	Specify sources (or clients) IP addresses used to match a forward policy rule	Source list specifies an IPv4 and/or IPv6 class list used to match a set of clients. Each entry in a source class list specifies an IPv4 or IPv6 address. Use the <code>match-class-list</code> command to assign one or more class lists. To match all source IP addresses, use the <code>match-any</code> command. If no source is present, <code>match-any</code> matches all sources, otherwise it will only be applied if the source does not fall under any other sources.
Destination Match Rule	Specifies a destination match rule, corresponding action to apply when there is a match, and the priority of the rule	Matches the destination URL or HOST string present in the HTTP request. The destination class list is of AC string type or IP type and may be part of a URL, HOST, or IP address. If the destination rule matches, the action specified in the rule is applied. The priority of the rule is used to break ties when multiple destination rules are matched. The higher priority value takes precedence. Use <code>destination any</code> to match all destinations if no other destination rule is present or if the destination does not fall under other rules. IP type class lists allows import and utilization of Threatstop files.
Action	Defines the action to take	Actions define what to do with the request and whether to log the event. The scope of the action

Table 17 : Configuration Resources for Explicit HTTP Proxy

Resource	Purpose	Description
	corresponding to the matching rule	object is within the forward policy. The action can be one of the following: <code>forward-to-internet</code>, <code>forward-to-service-group</code>, <code>forward-to-proxy</code>, <code>log</code>, or <code>drop</code>. Source NAT is an optional action that may be applied. In the case of an upstream proxy server and an SSLi configuration, <code>forward-to-proxy</code> is used instead of <code>forward-to-internet</code> or <code>forward-to-service-group</code> (refer to proxy chaining in this guide and the <i>SSL Configuration Guide</i> for more information).
Explicit HTTP-proxy port	Intercepts HTTP requests from clients	Virtual port that receives HTTP requests from traffic sources. This configuration resource consists of a real server configuration, a service group, a virtual server, and an HTTP virtual port. NOTE: If using <code>forward-to-internet</code> as a forwarding mechanism, you must configure an “Internet” real server and service-group. This can be thought of as a placeholder configuration which represents the “Internet” rather than an actual real device. This is an essential component to ensure feature functionality. Every allowed server port to the Internet must be configured in the real server and service group. For example, typically both port 80 and 443 are used for HTTP and HTTPS traffic. Both ports must be configured in the service group along with a real server representing the Internet.
Dynamic Service Template	Provides DNS resolution when forwarding to	Forward to Internet traffic requires DNS resolution. When the <code>forward-to-internet</code> command is used, you must define one or more DNS servers in a dynamic service template to

Table 17 : Configuration Resources for Explicit HTTP Proxy

Resource	Purpose	Description
	Internet	resolve host names to IP addresses.
Fallback Service Group		
Optionally, for any approved destinations which the ACOS Device cannot resolve, you can serve requests locally by configuring the following resources:		
Real Server (s) and service group	Locally serves content if destination cannot be reached	Standard SLB configuration with real servers and service group, to be called in Forward Policy
VIP with HTTP vPort	Receives requests to be served locally	
NAT Resources		
The following resources are required only if sources (clients) require source NAT.		
NAT pool	Assigns outside addresses to inside clients	Range of public (externally routable) IP addresses to assign to clients before forwarding the client traffic to the Internet.

Basic network resources, including network interface connections to the sources and destinations, and DNS servers, also are required.

Basic Transparent Proxy Configuration

A basic transparent proxy configuration requires a configuration to handle HTTP and HTTPs traffic separately, as HTTPS traffic differs from HTTP traffic in that the traffic is encrypted and requires certificates and keys to handle the back and forth communication. For more information on SSLi configuration, see the SSL Insight Configuration Guide.

The following configuration example shows the basic elements needed to configure a transparent proxy for the inside device and outside device. This example assumes a basic network has been set up.

CLI Configuration

Inside Device Configuration

A NAT pool is configured for source network address translation (SNAT).

```
ACOS(config)# ip nat pool NAT 203.0.113.5 203.0.113.5 netmask /32
```

An slb template is configured for DNS resolution so the traffic knows where to go to.

```
ACOS(config)# slb template dynamic-service DNS
ACOS(config-dynamic-service)# dns server 168.95.1.1
ACOS(config-dynamic-service)# exit
```

Now, the server to handle HTTP and HTTPs (SSL) traffic is configured as well as the service-groups which we bind the slb server to, through the two separate ports, 80 for HTTP traffic, and 8080 for HTTPs traffic.

```
ACOS(config)# slb server SSL 192.168.90.142
ACOS(config-real server)# health-check-disable
ACOS(config-real server)#port 80 tcp
ACOS(config-real server-node port)# health-check-disable
ACOS(config-real server-node port)# exit
ACOS(config-real server)#port 8080 tcp
ACOS(config-real server-node port)# health-check-disable
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
ACOS(config)# slb service-group HTTP tcp
ACOS(config-slb svc group)# health-check-disable
ACOS(config-slb svc group)# member SSL 80
ACOS(config-slb svc group-member:80)# exit
ACOS(config-slb svc group)# exit
ACOS(config)# slb service-group SSL_8080 tcp
ACOS(config-slb svc group)# health-check-disable
ACOS(config-slb svc group)# member SSL 8080
ACOS(config-slb svc group-member:8080)# exit
ACOS(config-slb svc group)# exit
```

Import the certificate and key, if they do not already reside on the device.

```
ACOS(config)# import cert ROOTCA.crt use-mgmt-port
tftp://192.168.1.25/ROOT-CA.crt
```

```

Done.
ACOS(config)# import key ROOTCA.key use-mgmt-port
tftp://192.168.1.25/ROOT-CA.key
Done

```

Next, a client-ssl template is configured to handle the certificate and key exchange for SSL (HTTPs) connections for transparent proxy. Note that `forward-proxy-enable` is only necessary for SSL Insight configuration to allow inspection through a man-in-the-middle inspection device.

```

ACOS(config)# slb template client-ssl SSL
ACOS(config-client ssl)# forward-proxy-ca-certificate ROOTCA.crt
ACOS(config-client ssl)# forward-proxy-enable
ACOS(config-client ssl)# exit

```

The HTTP policy (HTTP-policy) and HTTPs (SSL-policy) are configured where we simply take the appropriate traffic to its destination through the source and action sub-configurations. The source network address translation (SNAT), as an action, is an optional component.

```

ACOS(config)# slb template policy HTTP-POLICY
ACOS(config-policy)# forward-policy
ACOS(config-policy-forward-policy)# action A1
ACOS(config-policy-forward-policy-action)# forward-to-internet HTTP snat
NAT
ACOS(config-policy-forward-policy-action)# log
ACOS(config-policy-forward-policy-action)# exit
ACOS(config-policy-forward-policy)# source ANY
ACOS(config-policy-forward-policy-source)# match-any
ACOS(config-policy-forward-policy-source)# destination any action A1
ACOS(config-policy-forward-policy-source)# exit
ACOS(config-policy-forward-policy)# exit
ACOS(config-policy)# exit

ACOS(config)# slb template policy SSL-POLICY
ACOS(config-policy)# forward-policy
ACOS(config-policy-forward-policy)# action A1
ACOS(config-policy-forward-policy-action)# forward-to-internet SSL_8080
snat NAT
ACOS(config-policy-forward-policy-action)# exit

```

```

ACOS(config-policy-forward-policy)# source ANY
ACOS(config-policy-forward-policy-source)# match-any
ACOS(config-policy-forward-policy-source)# destination any action A1
ACOS(config-policy-forward-policy-source)# exit
ACOS(config-policy-forward-policy)# exit
ACOS(config-policy)# exit

```

Finally, we bind all these elements together in an slb virtual-server template using the wildcard VIP 0.0.0.0, with the HTTP and HTTPS traffic through the 80 and 443 virtual ports respectively.

```

ACOS(config)# slb virtual-server IN_VIP 0.0.0.0
ACOS(config-slb vserver)# port 80 http
ACOS(config-slb vserver-vport)# service-group HTTP
ACOS(config-slb vserver-vport)# template policy HTTP-POLICY
ACOS(config-slb vserver-vport)# template dynamic-service DNS
ACOS(config-slb vserver-vport)# exit
ACOS(config-slb vserver)# port 443 https
ACOS(config-slb vserver-vport)# service-group SSL_8080
ACOS(config-slb vserver-vport)# template policy SSL-POLICY
ACOS(config-slb vserver-vport)# template dynamic-service DNS
ACOS(config-slb vserver-vport)# template client-ssl SSL
ACOS(config-slb vserver-vport)# exit
ACOS(config-slb vserver)# exit

```

Outside Device Configuration

The configuration for the outside device (or partition), we need the following elements:

The slb server and service group are configured to receive HTTP and HTTPS traffic.

```

ACOS(config)# slb server DEFAULT_GATEWAY 20.1.1.10
ACOS(config-real server)# port 443 tcp
ACOS(config-real server-node port)# health-check-disable
ACOS(config-real server-node port)# exit
ACOS(config-real server)# port 0 tcp
ACOS(config-real server-node port)# health-check-disable
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit

ACOS(config)# slb service-group DG_SSL_SG tcp

```

```
ACOS(config-slb svc group)# member DEFAULT_GATEWAY 443
ACOS(config-slb svc group-member:443)# exit
ACOS(config-slb svc group)# exit
ACOS(config)# slb service-group DG_TCP_SG tcp
ACOS(config-slb svc group)# member DEFAULT_GATEWAY 0
ACOS(config-slb svc group-member:0)# exit
ACOS(config-slb svc group)# exit
```

The slb template server-ssl is configured for SSL Insight.

```
ACOS(config)# slb template server-ssl SSLINSIGHT_SERVERSIDE
ACOS(config-server ssl)# forward-proxy-enable
ACOS(config-server ssl)# exit
```

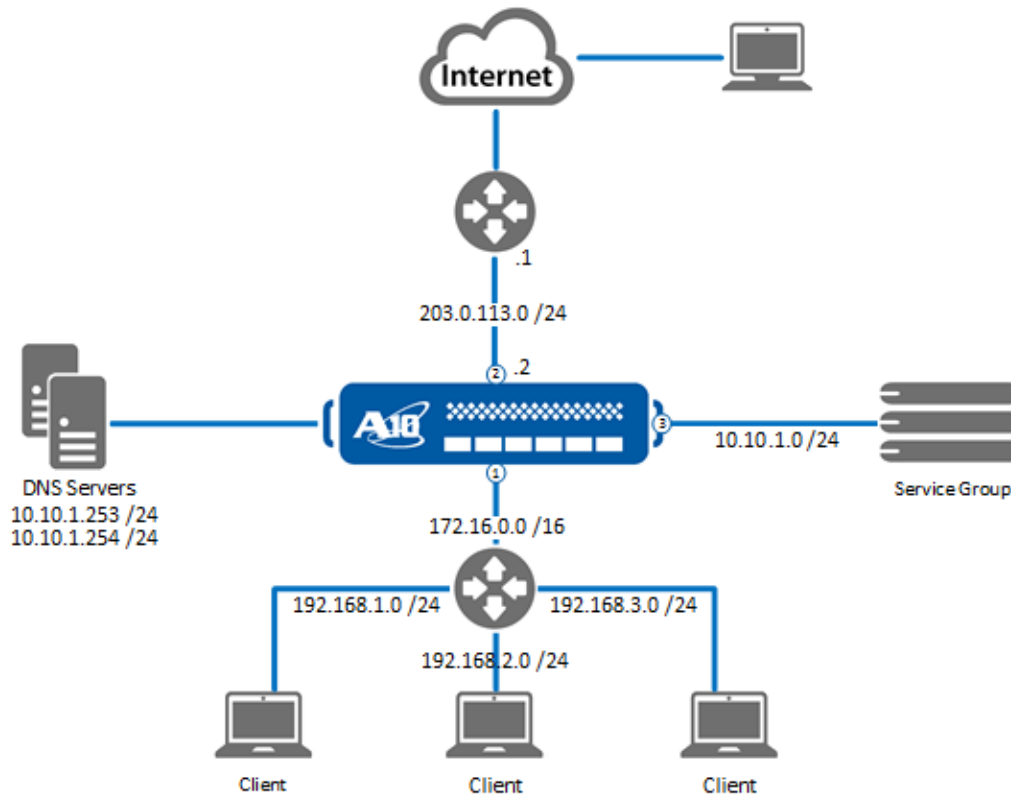
The virtual-server binds these components together for the HTTP and HTTPS traffic. The use-rcv-hop-for-resp is used to ensure correct direction of traffic.

```
ACOS(config)# slb virtual-server OUTSIDE_VIP 0.0.0.0
ACOS(config)# slb virtual-server OUTSIDE_VIP 20.1.1.20
ACOS(config-slb vserver)# port 8080 http
ACOS(config-slb vserver-vport)# no-dest-nat port-translation
ACOS(config-slb vserver-vport)# service-group DG_SSL_SG
ACOS(config-slb vserver-vport)# template server-ssl SSLINSIGHT_
SERVERSIDE
ACOS(config-slb vserver-vport)# use-rcv-hop-for-resp
ACOS(config-slb vserver-vport)# exit
ACOS(config-slb vserver)# port 0 tcp
ACOS(config-slb vserver-vport)# no-dest-nat
ACOS(config-slb vserver-vport)# service-group DG_TCP_SG
ACOS(config-slb vserver-vport)# use-rcv-hop-for-resp
ACOS(config-slb vserver-vport)# exit
ACOS(config-slb vserver)# exit
```

Basic Explicit Proxy Configuration

The basic configuration is to forward all traffic via Explicit Proxy to the Internet and log each request.

Figure 82 : Explicit HTTP Proxy Configuration



CLI Configuration

1. Configure network settings such as Ethernet data interfaces, VLANs, and routing so that there is connectivity between the traffic sources or clients, the destinations (Internet servers and internal real servers), and DNS servers, as shown in [Explicit HTTP Proxy Configuration](#).

```
ACOS(config)# interface ethernet 1
ACOS(config-if:ethernet:1)# ip address 10.10.1.254 255.255.255.0
ACOS(config-if:ethernet:1)# exit
ACOS(config)# interface ethernet 2
ACOS(config-if:ethernet:2)# ip address 203.0.113.254 255.255.255.0
ACOS(config-if:ethernet:2)# exit
ACOS(config)# interface ethernet 3
ACOS(config-if:ethernet:3)# ip address 172.168.0.254 255.255.255.0
ACOS(config-if:ethernet:3)# exit
```

```
ACOS(config)# ip route 0.0.0.0 /0 203.0.113.1
ACOS(config)# ip route 192.168.0.0 /16 172.16.0.1
```

2. Create a placeholder internal server and service group.

```
ACOS(config)# slb server Internet_Server 1.1.1.1
ACOS(config-real server)# port 80 tcp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# port 443 tcp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
ACOS(config)# slb service-group To_Internet tcp
ACOS(config-slb svc group)# member Internet_Server 80
ACOS(config-slb svc group-member:80)# exit
ACOS(config-slb svc group)# member Internet_Server 443
ACOS(config-slb svc group-member:443)# exit
ACOS(config-slb svc group)# exit
```

3. Create a dynamic service template and specify default Internet DNS resolvers.

```
ACOS(config)# slb template dynamic-service DNS
ACOS(config-dynamic-service)# dns server 10.10.1.253
ACOS(config-dynamic-service)# dns server 10.10.1.254
ACOS(config-dynamic-service)# exit
```

Alternatively, you can also specify a class-list with specific DNS server IPs for directing traffic to particular destination domains.

```
ACOS(config)# slb template dynamic-service DNS
ACOS(config-dynamic-service)# class-list test priority 20
ACOS(config-dynamic-service-class-list)# dns server 10.20.2.140
ACOS(config-dynamic-service-class-list)# dns server 10.20.20.420
```

4. Create a Source NAT Pool to use for Internet-bound traffic.

```
ACOS(config)# ip nat pool Internet_Pool 203.0.113.5 203.0.113.5 netmask /32
```

5. Create a Forward Policy.

```
ACOS(config)# slb template policy Explicit_Proxy
ACOS(config-policy)# forward-policy
```

6. Create an action for Internet-bound traffic to use the source NAT pool and log the

request.

```
ACOS(config-policy-forward-policy)# action Permit_to_Internet
ACOS(config-policy-forward-policy-action)# forward-to-internet To_Internet snat
Internet_Pool
ACOS(config-policy-forward-policy-action)# log
ACOS(config-policy-forward-policy-action)# exit
```

7. Create match-any source and destination rules.

```
ACOS(config-policy-forward-policy)# source Any_Source
ACOS(config-policy-forward-policy-source)# match-any
ACOS(config-policy-forward-policy-source)# destination any action Permit_to_
Internet
ACOS(config-policy-forward-policy-source)# exit
ACOS(config-policy-forward-policy)# exit
ACOS(config-policy)# exit
```

8. Create the explicit proxy virtual server to list on port 8080.

```
ACOS(config)# slb virtual-server EP_VIP 172.16.10.10
ACOS(config-slb vserver)# port 8080 http
ACOS(config-slb vserver-vport)# service-group To_Internet
ACOS(config-slb vserver-vport)# template policy Explicit_Proxy
ACOS(config-slb vserver-vport)# template dynamic-service DNS
ACOS(config-slb vserver-vport)# exit
ACOS(config-slb vserver)# exit
```

To make the basic explicit proxy configuration resilient, add a fallback service group to handle cases when a domain name cannot be resolved. All requests are then forwarded to the configured fallback service group.

9. Define fallback servers and a fallback service group.

```
ACOS(config)# slb server Fallback_Server1 10.10.1.20
ACOS(config-real server)# port 80 tcp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
ACOS(config)# slb server Fallback_Server2 10.10.1.21
ACOS(config-real server)# port 80 tcp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
```

```

ACOS(config)# slb service-group Fallback tcp
ACOS(config-slb svc group)# member Fallback_Server1 80
ACOS(config-slb svc group-member:80)# exit
ACOS(config-slb svc group)# member Fallback_Server2 80
ACOS(config-slb svc group-member:80)# exit
ACOS(config-slb svc group)# exit

```

10. Add a different source NAT pool for forwarded requests to the fallback service group.

```

ACOS(config)# ip nat pool Fallback_Pool 10.10.1.120 10.10.1.120 netmask /24

```

11. Add fallback service group to the action in the existing SLB template policy, Explicit_Proxy using step 5 first then the following commands:

```

ACOS(config)# slb template policy Explicit_Proxy
ACOS(config-policy)# forward-policy
ACOS(config-policy-forward-policy)# action Permit_to_Internet
ACOS(config-policy-forward-policy-action)# forward-to-internet To_Internet
snat Internet_Pool fallback Fallback snat Fallback_Pool
ACOS(config-policy-forward-policy-action)# log
ACOS(config-policy-forward-policy-action)# exit
ACOS(config-policy-forward-policy)# exit
ACOS(config-policy)# exit

```

GUI Configuration

Configure network settings such as Ethernet data interfaces, VLANs, and routing so that there is connectivity between the traffic sources or clients, the destinations (Internet servers and internal real servers), and DNS servers, as shown in [Explicit HTTP Proxy Configuration](#). This example assumes three ethernet interface connections are established.

1. Define ethernet interface settings.
2. Navigate to **Network >> Interfaces**.
3. Click **Edit** for e1 Interface.
4. Click **+** in the IP section.
5. In the IP Address field, enter "10.10.1.254".
6. In the Netmask field, enter "255.255.255.0".

7. Click **Add**.
8. Click **Update**.
9. Click **Edit** for e2 Interface.
10. Click **+** in the IP section.
11. In the IP Address field, enter "203.0.113.254".
12. In the Netmask field, enter "255.255.255.0".
13. Click **Add**.
14. Click **Update**.
15. Click **Edit** for e3 Interface.
16. Click **+** in the IP section.
17. In the IP Address field, enter "172.168.0.254".
18. In the Netmask field, enter "255.255.255.0".
19. Click **Add**.
20. Click **Update**.

Establish IP Routes.

1. Navigate to **Network >> Routes**.
2. Click **+** **Create**.
3. In the IP Dest Address field, enter "0.0.0.0".
4. In the IP Mask field, enter "/0".
5. In the IP Next Hop field, enter "203.0.113.1".
6. Click **Add**.
7. Click **Create Route**.
8. Click **+** **Create**.
9. In the IP Dest Address field, enter "192.168.0.0".
10. In the IP Mask field, enter "/16".
11. In the IP Next Hop field, enter "172.16.160.1".

12. Click **Add**.

13. Click **Create Route**.

Create a placeholder internal server.

1. Navigate to **Security >> Forward Proxy >> Servers** tab.
2. Click **+ Create**.
3. In the Name field, enter "Internet_Server".
4. In the Host field, enter "1.1.1.1".
5. Click **+ Add Port**.
6. In the Port field, enter "80" and click Apply.
7. Click **+ Add Port**.
8. In the Port field, enter "443" and click Apply.
9. Click **OK**.

Create our virtual server to list on port 8080.

1. Navigate to **Security >> Forward Proxy >> Services** tab.
2. Click **+ Create**.
3. In the Name field, enter "EP_VIP".
4. In the IPv4 Address field, enter "172.16.10.10".
5. Click **+ Add Port**.
6. In the Port field, enter "8080".
7. In the Protocol field, select HTTP from the drop-down menu.
8. Define our Service Group and our server as a member for ports 80 and 443.
9. Click **+** in the Service Group section.
10. In the Name field, enter "To_Internet".
11. Click **+ Add Member**.
12. In the Name field, click the drop-down menu and click **Internet_Server**.
13. In the Port field, enter "80" and click **Apply**.
14. Click **+ Add Member**.

15. In the Name field, click the drop-down menu and click **Internet_Server**.
16. In the Port field, enter “443” and click **Apply**.
17. Click **OK**.

Create our internet DNS resolvers by creating a dynamic service template.

1. Click + in the Dynamic Service section.
2. In the Name field, enter “dynamic_service”
3. In the IPv4 DNS Server 1 field, enter “10.10.1.253”.
4. In the IPv4 DNS Server 2 field, enter “10.10.1.254”.
5. Click **OK**.

Create a Source NAT Pool to use for Internet-bound traffic.

1. Click + in the SNAT Pool section.
2. In the Name field, enter “Internet_Pool”.
3. In the Start Address field, enter “203.0.113.5”.
4. In the End Address field, enter “203.0.113.5”.
5. In the Netmask field, enter “/32”.
6. Click **OK**.

Create a Forward Policy, and the action policy and source policy to be followed.

1. Click + in the Policy Template section.
2. In the Name field, enter “Explicit_Proxy”.
3. Click the **Action Policies** tab.
4. Click + **Add**.
5. In the Name field, enter “Permit_to_Internet”.
6. In the Action field, click the drop-down menu, select **Forward Request to Internet**.
7. In the Service Group field, click the drop-down menu, and click **To_Internet**.
8. In the IPv4 Pool field, click the drop-down menu, and click **Internet_Pool**.
9. Click the check box next to the Log field to enable logging.

10. Click **OK**.
11. Click the **Source Policies** tab.
12. Click **+ Add**.
13. In the Name field, enter “Any_Source”.
14. In the Match Type field, click the drop-down menu and click **Any**.
15. Click **+ Add**.
16. Under Destination Rules, in Match Type, click the drop-down menu and click **Any**.
17. In the Action field, click the drop-down menu and click **Permit_to_Internet**.
18. Click **Apply**.
19. Click **OK**.
20. Click **Add Template**.

Now, we will make our configuration more resilient, by creating a fallback service group to handle cases when a domain name cannot be resolved. All requests will then be forwarded to the configured fallback service group. Take the following steps:

1. Navigate to **Security >> Forward Proxy >> Servers** tab.
2. Click **+ Create**.
3. In the Name field, enter “Fallback_Server1”.
4. In the Host field, enter “10.10.1.20”.
5. Click **+ Add Port**.
6. In the Port field, enter “80” and click **Apply**.
7. Click **OK**.
8. Click **+ Create**.
9. In the Name field, enter “Fallback_Server2”.
10. In the Host field, enter “10.10.1.21”.
11. Click **+ Add Port**.
12. In the Port field, enter “80” and click **Apply**.
13. Click **OK**.
14. Click **Service Groups**.

15. Click **+ Create**.
16. In the Name field, enter Fallback.
17. Click **+ Add Member**.
18. In the Name field, click the drop-down menu and click **Fallback_Server1**.
19. In the Port field, enter “80” and click **Apply**.
20. Click **+ Add Member**.
21. In the Name field, click the drop-down menu and click **Fallback_Server1**.
22. In the Port field, enter “80” and click **Apply**.
23. Click **OK**.

After creating the fallback servers and service group, edit our existing forward policy.

1. Click the **Templates** tab.
2. Click **Edit** for Explicit_Proxy.
3. Click **Action Policies tab** if not already there.
4. Click **Edit** for Permit_to_Internet.
5. In the Fallback SG field, click the drop-down menu and click **Fallback**.
6. In the Fallback IPv4 Pool field, click **+**.
7. In the Name field, enter “Fallback_Pool”.
8. In the Start Address field, enter “10.10.1.120”.
9. In the End Address field, enter “10.10.1.139”.
10. In the Netmask field, enter “/24”.
11. Click **OK**.
12. Click **Update**.

Advanced Explicit Proxy Configuration

This configuration example demonstrates the use of specific allowed sources and target destinations accessing the Internet. An internal service group for providing a software update service is configured, and denies access to others. This configuration

builds upon the basic explicit proxy configuration in the previous section and follows the topology from [Explicit HTTP Proxy Configuration](#).

CLI Configuration

1. Create a source class list of IPv4 type. Source class lists may be of IPv4 or IPv6 types and are defined as a list of IP subnets and network masks.

```
ACOS(config)# class-list Corporate_Clients ipv4
ACOS(config-class list)# 192.168.1.0/24
ACOS(config-class list)# 192.168.2.0/26
ACOS(config-class list)# 172.16.0.0/16
ACOS(config-class list)# exit
ACOS(config)# class-list Corporate_Servers ipv4
ACOS(config-class list)# 10.10.1.10/32
ACOS(config-class list)# 10.10.1.11/32
ACOS(config-class list)# 10.10.1.12/32
ACOS(config-class list)# exit
```

2. Create destination class lists of Aho-Corasick string type or IP type. For Aho-Corasick string type, destinations specified may be partial strings matching the HOST field or URL field of an HTTP request. String match is not case-sensitive.

```
ACOS(config)# class-list Allowed_Destinations ac
ACOS(config-class list)# contains yahoo
ACOS(config-class list)# contains google
ACOS(config-class list)# contains facebook
ACOS(config-class list)# contains cnn
ACOS(config-class list)# contains disney
ACOS(config-class list)# contains ebay
ACOS(config-class list)# contains bankofamerica
ACOS(config-class list)# exit
ACOS(config)# class-list Restricted_Destinations ac
ACOS(config-class list)# contains youtube
ACOS(config-class list)# contains netflix
ACOS(config-class list)# contains hulu
ACOS(config-class list)# exit
ACOS(config)# class-list Internal_Destinations ac
ACOS(config-class list)# contains acme-inc
ACOS(config-class list)# exit
ACOS(config)# class-list Allowed_Client-urls ac
```

```
ACOS(config-class list)# contains news
ACOS(config-class list)# contains finance
ACOS(config-class list)# contains sports
ACOS(config-class list)# contains entertainment
ACOS(config-class list)# exit
ACOS(config)# class-list Update_Destinations ac
ACOS(config-class list)# contains update.microsoft
ACOS(config-class list)# contains download.microsoft
ACOS(config-class list)# contains windowsupdate
ACOS(config-class list)# contains archive.linux
ACOS(config-class list)# contains archive.ubuntu
ACOS(config-class list)# contains mirror.ubuntu
ACOS(config-class list)# starts-with 10.10.10.200
ACOS(config-class list)# exit
ACOS(config)# class-list Forbidden_Destination ipv4
ACOS(config-class list)# 198.51.100.0/24
ACOS(config-class list)# exit
```

3. Configure software update servers, service groups, and NAT pool to handle forward-to-service-group and permit-to-internet requests.

```
ACOS(config)# slb server Update_Server1 10.10.1.31
ACOS(config-real server)# port 80 tcp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
ACOS(config)# slb server Update_Server2 10.10.1.32
ACOS(config-real server)# port 80 tcp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
ACOS(config)# slb server Internet_Server 1.1.1.1
ACOS(config-real server)# port 80 tcp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# port 443 tcp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
ACOS(config)# slb service-group Updates tcp
ACOS(config-slb svc group)# member Update_Server1 80
ACOS(config-slb svc group-member:80)# exit
ACOS(config-slb svc group)# member Update_Server2 80
ACOS(config-slb svc group-member:80)# exit
```

```

ACOS(config-slb svc group)# exit
ACOS(config)# slb service-group To_Internet tcp
ACOS(config-slb svc group)# member Internet_Server 80
ACOS(config-slb svc group-member:80)# exit
ACOS(config-slb svc group)# member Internet_Server 443
ACOS(config-slb svc group-member:443)# exit
ACOS(config-slb svc group)# exit
ACOS(config)# ip nat pool Internet_Pool 203.0.113.5 203.0.113.5 netmask
/32

```

4. Configure forward-to-service-group, Permit-to-Internet, and drop actions.

```

ACOS(config)# slb template policy Explicit_Proxy
ACOS(config-policy)# forward-policy
ACOS(config-policy-forward-policy)# action Default_Deny
ACOS(config-policy-forward-policy-action)# drop
ACOS(config-policy-forward-policy-action)# drop-message "Access is
restricted."
ACOS(config-policy-forward-policy-action)# log
ACOS(config-policy-forward-policy-action)# exit
ACOS(config-policy-forward-policy)# action Permit_Software-Updates
ACOS(config-policy-forward-policy-action)# forward-to-service-group
Updates
ACOS(config-policy-forward-policy-action)# log
ACOS(config-policy-forward-policy-action)# exit
ACOS(config-policy-forward-policy)# action Permit_To_Internet
ACOS(config-policy-forward-policy-action)# forward-to-internet To_
Internet snat Internet_Pool
ACOS(config-policy-forward-policy-action)# log
ACOS(config-policy-forward-policy-action)# exit

```

5. Modify the source rule to match-any and configure destination to match any for dropping and logging.

```

ACOS(config-policy-forward-policy)# source Any_Source
ACOS(config-policy-forward-policy-source)# match-any
ACOS(config-policy-forward-policy-source)# destination any action
Default_Deny
ACOS(config-policy-forward-policy-source)# exit

```

6. Configure a source rule and add up to two source class lists as a single match rule

to a forward policy. Multiple source match rules may be configured per forward policy using the OR clause.

```
ACOS(config-policy-forward-policy)# source Allowed_Clients
ACOS(config-policy-forward-policy-source)# match-class-list Corporate_
Clients
ACOS(config-policy-forward-policy-source)# exit
ACOS(config-policy-forward-policy)# source Allowed_Servers
ACOS(config-policy-forward-policy-source)# match-class-list Corporate_
Servers
ACOS(config-policy-forward-policy-source)# exit
```

7. Add one or more destination class lists as matching rules for a specific source class rule. Up to 1024 destination rules may be added to a single source match rule. The default action for non-matching destination match rules is to drop the request.

```
ACOS(config-policy-forward-policy)# source Allowed_Clients
ACOS(config-policy-forward-policy-source)# match-class-list Corporate_
Clients
ACOS(config-policy-forward-policy-source)# destination class-list
Allowed_Destinations action Permit_To_Internet host priority 100
ACOS(config-policy-forward-policy-source)# destination class-list
Internal_Destinations action Permit_Software-Updates host priority 200
ACOS(config-policy-forward-policy-source)# destination class-list
Allowed_Client-urls action Permit_To_Internet url priority 300
ACOS(config-policy-forward-policy-source)# destination class-list
Update_Destinations action Permit_To_Internet host priority 400
ACOS(config-policy-forward-policy-source)# destination class-list
Restricted_Destinations action Default_Deny host priority 500
ACOS(config-policy-forward-policy-source)# destination class-list
Forbidden_Destination action Default_Deny ip priority 600
ACOS(config-policy-forward-policy-source)# exit
ACOS(config-policy-forward-policy)# source Allowed_Servers
ACOS(config-policy-forward-policy-source)# match-class-list Corporate_
Servers
ACOS(config-policy-forward-policy-source)# destination class-list
Update_Destinations action Permit_To_Internet host priority 100
ACOS(config-policy-forward-policy-source)# exit
ACOS(config-policy-forward-policy)# exit
```

```
ACOS(config-policy)# exit
```

For creating a forward policy, go to the forward-policy sub-command for configuration of the various types of templates. Templates and parameters are elaborated upon here.

- Source Rule (source) – Defines the match rule for the traffic sources. The source sub-command within forward policy names the source rule. Multiple source rules may be defined in a forward policy. But only a single source rule of match-any may be defined for a forward policy. IP addresses in multiple configured source class-lists configured for a forward policy cannot overlap.
 - match-class-list – IPv4 or IPv6 class list that specifies hosts or subnets for matching source rules.
 - match-any - default match rule if no other specific class list is matched.
- Destination Rule (destination)– The destination rule is associated with a source rule and defines either a specific class-list or default match-any. A unique destination class for a particular source rule.
- Destination Class-List (class-list) – Class list that contains the URL or hostname strings or IP range for destinations that clients are allowed to access.
- Destination Web-Category-List (web-category-list) - Category list that contains predefined website URLs for destinations that clients are allowed to access.
 - priority – This is used to break ties when there are multiple destination rules that match the traffic such that only a single destination rule’s action is applied to the traffic. The higher value takes priority over the lower value. The priority value is unique per source rule.
 - action – Defines where to forward the packet upon receipt of request and whether to log the event.
- Action type – Defines how traffic will be forwarded, can be `drop`, `forward-to-internet`, `forward-to-service-group` or `forward-to-proxy`. Note that these are mutually exclusive actions. Multiple actions may be configured under a single policy. Forward-to-proxy should be used in the case of an upstream proxy server to ensure proper proxy information is not lost during communication.
- Drop message (`drop-message`) - When a request is dropped, a message is displayed. A default “Access to this site is blocked by administrator” message appears if no text is provided.

- Drop and redirect url (`drop-redirect-url`) - When a request is dropped, the client request is redirected to another specified URL. For the `http-status-code`, the default is 302 Found.
- Source NAT (`snat`) – Previously configured NAT pool. This is an optional component.
- Fallback (`fallback`) – Forwards traffic to a fallback service group if the ACOS Device is unable to reach the destination and relies on a previously configured service group.
- Log (`log`) – HTTP requests forwarded or dropped can be logged to the external and/or internal syslog servers using this command.
- No Client Connection Reuse (`no-client-conn-reuse`) – Configure this option only when the HTTP/HTTPS client will not send multiple requests to different destinations over the same TCP connection between the client and the ACOS Device. Most modern web browsers have connection reuse enabled by default thus this setting typically doesn't apply when clients are web browsers. In this mode, the explicit proxy inspects only the first HTTP request after a TCP connection is established and applies the forward policy. All subsequent requests on that TCP connection are not inspected and are tunneled directly to the same destination. This mode is designed for higher performance.

The web-category license file must be imported and then `enable` must be invoked prior to the use of category-list within the web-category sub-configuration.

Commands `drop-message` and `drop-redirect-url` are mutually exclusive actions. If both are entered, the prior command will be overwritten by the more recent one.

GUI Configuration

This configuration builds upon the basic explicit proxy configuration in the previous section and follows the topology from [Explicit HTTP Proxy Configuration](#). Take the following steps:

Create a source class list of IPv4 type. Source class lists may be of IPv4 or IPv6 types and are defined as a list of IP subnets and network masks.

1. Navigate to **Security>> Forward Proxy**.
2. Click **Class List** tab, and click **Configuration**.

3. Click **+ Create**.
4. In the Name field, enter "Corporate_Clients".
5. In the IPv4 Address field, enter "192.168.1.0/24" and click **Add**.
6. Go back to IPv4 Address field, enter "192.168.2.0/24" and click **Add**.
7. Go back to IPv4 Address field, enter "172.16.0.0/16" and click **Add**.
8. Click **OK**.
9. Click **+ Create**.
10. In the Name field, enter "Corporate_Servers".
11. In the IPv4 Address field, enter "10.10.1.10/32" and click **Add**.
12. Go back to IPv4 Address field, enter "10.10.1.11/32" and click **Add**.
13. Go back to IPv4 Address field, enter "10.10.1.12/32" and click **Add**.
14. Click **OK**.

Create destination class lists of Aho-Corasick string type. Destinations specified may be partial strings matching the HOST field or URL field of an HTTP request. String match is not case-sensitive.

1. Ensure you are in **Security>> Forward Proxy>> Class Lists**. If not, follow steps 1 and 2 in the prior section.
2. Click **+ Create**.
3. In the Name field, enter "Allowed_Destinations".
4. In the Type field, click **Aho Corasick**.
5. In AC (Aho Corasick) empty field, enter "yahoo" and click **Add**.
6. Go back to this field, enter "google" and click **Add**.
7. Go back to this field, enter "facebook" and click **Add**.
8. Go back to this field, enter "cnn" and click **Add**.
9. Go back to this field, enter "disney" and click **Add**.
10. Go back to this field, enter "ebay" and click **Add**.
11. Go back to this field, enter "bankofamerica" and click **Add**.
12. Click **OK**.

13. Click **+ Create**.
14. In the Name field, enter “Restricted_Destinations”.
15. In the Type field, click **Aho Corasick**.
16. In AC (Aho Corasick) empty field, enter “youtube” and click **Add**.
17. Go back to this field, enter “netflix” and click **Add**.
18. Go back to this field, enter “hulu” and click **Add**.
19. Go back to this field, enter “cnn” and click **Add**.
20. Go back to this field, enter “disney” and click **Add**.
21. Go back to this field, enter “ebay” and click **Add**.
22. Go back to this field, enter “bankofamerica” and click **Add**.
23. Click **OK**.
24. Click **+ Create**.
25. In the Name field, enter “Internal_Destinations”.
26. In the Type field, click **Aho Corasick**.
27. In AC (Aho Corasick) empty field, enter “acme-inc” and click **Add**.
28. Click **OK**.
29. Click **+ Create**.
30. In the Name field, enter “Allowed_Client-urls”.
31. In the Type field, click **Aho Corasick**.
32. In AC (Aho Corasick) empty field, enter “news” and click **Add**.
33. Go back to this field, enter “finance” and click **Add**.
34. Go back to this field, enter “sports” and click **Add**.
35. Go back to this field, enter “entertainment” and click **Add**.
36. Click **OK**.
37. Click **+ Create**.
38. In the Name field, enter “Update_Destinations”.
39. In the Type field, click **Aho Corasick**.

40. In AC (Aho Corasick) empty field, enter “update.microsoft” and click **Add**.
41. Go back to this field, enter “download.microsoft” and click **Add**.
42. Go back to this field, enter “windowsupdate” and click **Add**.
43. Go back to this field, enter “archive.linux” and click **Add**.
44. Go back to this field, enter “archive.ubuntu” and click **Add**.
45. Go back to this field, enter “mirror.ubuntu” and click **Add**.
46. Click the Type drop-down menu, and click **Starts With**.
47. In AC (Aho Corasick) empty field, enter “10.10.10.200” and click **Add**.
48. Click **OK**.

Create a destination IP type class list for use.

1. Click **+ Create**.
2. In the Name field, enter “Forbidden_Destinations”.
3. In the Type field, click **IPv4**.
4. In the IPv4 Address field, enter “198.51.100.0/24” and click **Add**.
5. Click **OK**.

Configure software update servers and service group to handle forward-to-service-group requests.

1. Click **Servers** tab.
2. Click **+ Create**.
3. In the Name field, enter “Update_Server1”.
4. In the Host field, enter “10.10.1.31”.
5. Click **+ Add Port**.
6. In the Port field, enter “80” and click **Apply**.
7. Click **OK**.
8. Click **+ Create**.
9. In the Name field, enter “Update_Server2”.
10. In the Host field, enter “10.10.1.32”.

11. Click **+ Add Port**.
12. In the Port field, enter “80” and click **Apply**.
13. Click **OK**.

Create our service group for the update servers we created.

1. Click **Service Groups** tab.
2. Click **+ Create**.
3. In the Name field, enter “Updates”.
4. Click **+ Add Member**.
5. Click the **Name** drop-down menu and click Update_Server1.
6. In the Port field, enter “80” and click **Apply**.
7. Click **+ Add Member**.
8. Click the **Name** drop-down menu and click **Update_Server2**.
9. In the Port field, enter “80” and click **Apply**.
10. Click **OK**.

Edit our Explicit_Proxy policy with new actions.

1. Click **Templates** tab.
2. Click **Edit** for our existing Explicit_Proxy policy.
3. Under the Action Policies tab, click **+ Add**.
4. In the Name field, enter “Default-Deny”.
5. Click the **Action** drop-down menu, and Click **Drop**.
6. In the Drop Message field, enter “Access is restricted.”
7. Click the check box for Logging.
8. Click **OK**.
9. Click **Update**.
10. Click **Edit** for our existing Explicit_Proxy policy.
11. Click **+ Add** to create another action.
12. In the Name field, enter “Permit_Software-Updates”.

13. Click the **Action** drop-down menu, and Click **Forward Request to Service Group**.
14. Click the **Service Group** drop-down menu, and click **Updates**.
15. Click the check box for Logging.
16. Click **OK**.
17. Click **Update**.

Next, modify our existing source policy to add our new action policy, Default_Deny.

1. Click **Source Policies** tab.
2. Click **Edit** for Any_Source.
3. Click **Edit** for our Destination Rules Any.
4. Click the **Action** drop-down menu, click **Default_Deny**, and click **Apply**.
5. Click **Update**.

Next, create destination rules under a new source policy for our created Corporate_Clients.

1. Click **+ Add**.
2. In the Name field, enter "Allowed_Clients".
3. Click the **Match Type** drop-down menu and click **Class List**.
4. Click the **Class List** drop-down menu and click **Corporate_Clients**.
5. In Destination Rules, click **+ Add**.
6. Click the **Destination Rules Match Type** drop-down menu, and click **Class List**.
7. Click the **Value** drop-down menu, and click **Allowed_Destinations**.
8. Click the **Action** drop-down menu, and click **Permit_to_Internet**.
9. Click the **Match** drop-down menu, and click **Host**.
10. In the Priority field, enter "100".
11. Click **Apply**.
12. In Destination Rules, click **+ Add**.
13. Click the **Destination Rules Match Type** drop-down menu, and click **Class List**.
14. Click the **Value** drop-down menu, and click **Internal_Destinations**.

15. Click the **Action** drop-down menu, and click **Permit_Software-Updates**.
16. Click the **Match** drop-down menu, and click **Host**.
17. In the Priority field, enter “200”.
18. Click **Apply**.
19. In Destination Rules, click **+ Add**.
20. Click the **Destination Rules Match Type** drop-down menu, and click **Class List**.
21. Click the **Value** drop-down menu, and click **Allowed_Clients-urls**.
22. Click the **Action** drop-down menu, and click **Permit_to_Internet**.
23. Click the **Match** drop-down menu, and click **URL**.
24. In the Priority field, enter “300”.
25. Click **Apply**.
26. In Destination Rules, click **+ Add**.
27. Click the **Destination Rules Match Type** drop-down menu, and click **Class List**.
28. Click the **Value** drop-down menu, and click **Update_Destinations**.
29. Click the **Action** drop-down menu, and click **Permit_to_Internet**.
30. Click the **Match** drop-down menu, and click **Host**.
31. In the Priority field, enter “400”.
32. Click **Apply**.
33. In Destination Rules, click **+ Add**.
34. Click the **Destination Rules Match Type** drop-down menu, and click **Class List**.
35. Click the Value drop-down menu, and click **Restricted_Destinations**.
36. Click the Action drop-down menu, and click **Default-Deny**.
37. Click the Match drop-down menu, and click **Host**.
38. In the Priority field, enter “500”.
39. Click **Apply**.
40. Click **Update**.
41. In Destination Rules, click **+ Add**.

42. Click the **Destination Rules Match Type** drop-down menu, and click **Class List**.
43. Click the **Value** drop-down menu, and click **Forbidden_Destinations**.
44. Click the **Action** drop-down menu, and click **Default-Deny**.
45. Click the **Match** drop-down menu, and click **IP**.
46. In the Priority field, enter “600”.
47. Click **Apply**.
48. Click **Update**.

Next, we create a source policy for our Corporate_Servers and add a destination rule.

1. Click **+ Add**.
2. In the Name field, enter “Allowed_Servers”.
3. Click the **Match Type** drop-down menu and click **Class List**.
4. Click the **Class List** drop-down menu and click **Corporate_Servers**.
5. In Destination Rules, click **+ Add**.
6. Click the **Destination Rules Match Type** drop-down menu, and click **Class List**.
7. Click the **Value** drop-down menu, and click **Update_Destinations**.
8. Click the **Action** drop-down menu, and click **Permit_to_Internet**.
9. Click the **Match** drop-down menu, and click **Host**.
10. In the Priority field, enter “100”.
11. Click **Apply**.
12. Click **OK**.

Explicit Proxy URL Filtering

In addition to using a destination class-list, an object called category-list is available through the web-category configuration/feature. This allows control over an entire web category rather than having to enter every website or URL. Defined category-lists and updated policies are provided below to illustrate the use of url filtering. This example builds upon the advanced explicit proxy configuration in the initial section and assumes the library for web-category has been enabled.

CLI Configuration

1. Create a category-list from the web-category configuration for use as, in addition to, or as a replacement for destination class lists. This supplements or replaces destination class lists of Aho-Corasick string type or IP type in Advanced Explicit Proxy Configuration (CLI Example).

```
ACOS(config)# web-category
ACOS(config-web-category)# enable
ACOS(config-web-category)# category-list Finance_Sites
ACOS(config-web-category-category-list)# financial-services
ACOS(config-web-category-category-list)# business-and-economy
ACOS(config-web-category-category-list)# exit
ACOS(config-web-category)# category-list Search_Sites
ACOS(config-web-category-category-list)# search-engines
ACOS(config-web-category-category-list)# internet-portals
ACOS(config-web-category-category-list)# exit
ACOS(config-web-category)# exit
```

2. Confirm that the web-category feature is enabled using the **show web-category license** command.
3. Display the web-category-list configurations with the **show running-config web-category** command:

```
ACOS(config)# show running-config web-category
!Section configuration: 169 bytes
!
web-category
  category-list Finance_Sites
    financial-services
    business-and-economy
  category-list Search_Sites
    search-engines
    internet-portals
```

4. Similar to destination class lists, destination web-category-lists can be used in the source class rule, either supplementing or replacing them. Below is a modified version that replaces the source configuration in destination class lists as matching rules for a specific source class rule of the Advanced Explicit Proxy Configuration (CLI Example) example where we replace the destination class-list

rules with the destination web-category-list rules.

```
ACOS(config-policy-forward-policy)# source Allowed_Clients
ACOS(config-policy-forward-policy-source)# match-class-list Corporate_
Clients
ACOS(config-policy-forward-policy-source)# destination web-category-
list Finance_Sites action Permit_To_Internet host priority 100
ACOS(config-policy-forward-policy-source)# destination class-list
Internal_Destinations action Permit_Software-Updates host priority 200
ACOS(config-policy-forward-policy-source)# destination class-list
Allowed_Client-urls action Permit_To_Internet url priority 300
ACOS(config-policy-forward-policy-source)# destination class-list
Update_Destinations action Permit_To_Internet host priority 400
ACOS(config-policy-forward-policy-source)# destination web-category-
list Search_Sites action Default_Deny host priority 500
ACOS(config-policy-forward-policy-source)# destination class-list
Forbidden_Destination action Default_Deny ip priority 600
ACOS(config-policy-forward-policy-source)# exit
ACOS(config-policy-forward-policy)# source Allowed_Servers
ACOS(config-policy-forward-policy-source)# match-class-list Corporate_
Servers
ACOS(config-policy-forward-policy-source)# destination class-list
Update_Destinations action Permit_To_Internet host priority 100
ACOS(config-policy-forward-policy-source)# exit
ACOS(config-policy-forward-policy)# exit
ACOS(config-policy)# exit
```

GUI Configuration

Create a category list from pre-existing lists under web-category. These lists can be used in addition to, or as a replacement for destination class lists shown in the [GUI Configuration](#). A valid webroot license is required, and web categories must be enabled for this feature to work.

An example for creating two category lists follow. Take the following steps:

1. Navigate to **Security>> Web Categories**.
2. In the **Category List** section, click **+ Create**.
3. In the Name field, enter "Finance_Sites".

4. In the **Categories** section, click the business-and-economy and financial-services check boxes.
5. Click **OK**.
6. Click **Update**.
7. In the Category List section, click **+ Create**.
8. In the Name field, enter "Search_Sites".
9. In the **Categories** section, click the internet-portals and search-engines check boxes.
10. Click **OK**.
11. Click **Update**.

Similar to destination class lists, category-lists from Web Categories can be used in the source class rule, either supplementing or replacing them. For a source policy destination rule, to use a category list from web category, choose Web Category for the Destination Rules Match Type instead of Class List. Below is an example showing the steps to modify the destination rules for source policy Allowed_Clients in the Advanced Explicit Configuration GUI example of existing destination class lists with category lists.

1. Navigate to **Security>> Forward Proxy >> Templates** tab.
2. Click **Edit** for Explicit_Proxy policy.
3. Click **Source Policies** tab.
4. Click **Edit** for Allowed_Clients.
5. Click **Edit** for the Destination Rules with a Priority of 100.
6. Click the **Destination Rules Match Type** drop-down menu, and click **Web Category**.
7. Click the **Value** drop-down menu and click **Finance_Sites**.
8. Click **Apply**.
9. Click **Edit** for the **Destination Rules** with a Priority of 500.
10. Click the **Destination Rules Match Type** drop-down menu, and click **Web Category**.
11. Click the **Value** drop-down menu and click **Search_Sites**.

12. Click **Apply**.
13. Click **Update**.

Transparent Proxy URL Filtering

ACOS supports configuring following types of URL filtering in transparent proxy mode:

- URL filtering based on Subject Name Indicator (SNI) without a client-SSL template
- URL filtering based on Subject Alternative Name (SAN) with a client-SSL template

URL Filtering Based on SNI

The URL filtering can be performed based on SNI extracted from the client Hello message, if present. By default, the SNI-based filtering is enabled. To configure SNI-based filtering, bind a template policy to a wildcard VIP HTTPS virtual port. The virtual port must not have a client-ssl template configured. Both IPv4 and IPv6 addressing is supported for this feature.

Currently, this type of SNI-based filtering is supported only for transparent proxy configurations where the virtual port is of type HTTPS. Only the HTTPS traffic designated for the configured virtual port is inspected. The HTTPS traffic flowing through other ports is not inspected.

The following lists the supported configurations for HTTPS vPort in terms of having a template policy and a client-SSL template configured:

- If the virtual port has a template policy configured, SNI-based filtering is configured. Policy match is based on the SNI.
- If the virtual port has both a template policy and a client-ssl template, the configuration is switched to transparent proxy and SSLi.
- If virtual port has only a client-ssl template, the configuration is switched to SSLi.
- If there is no SNI extension present, the connection is dropped. However, if one of these options is set, then no SNI session can pass:
 - `ssli-url-filtering no-sni-allow`
 - `ssli-url-filtering bypassed-sni-disable`

You can check the SNI-based filtering statistics using the `show slb template policy policy_name forward-policy-stats` command. For more information about the command syntax, see the *CLI Reference Guide for ADC*.

CLI Configuration

The following procedure configures an ACOS device in transparent proxy mode to perform SNI-based filtering of traffic.

1. Configure the real server called `gateway` at IP address `10.10.10.2`. Port `80` is added for TCP traffic.

```
ACOS(config)# slb server gateway 10.10.10.2
ACOS(config-real server)# port 80 tcp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
```

2. Configure a TCP service group called `gateway_http` and associate the real server `gateway` and port `80` to the service group.

```
ACOS(config)# slb service-group gateway_http tcp
ACOS(config-slb svc group)# member gateway 80
ACOS(config-slb svc group-member:80)# exit
ACOS(config-slb svc group)# exit
```

3. Configure a template policy called `test`. Enable `forward-policy` to specify permitted traffic destinations and sources along with the actions to apply.

```
ACOS(config)# slb template policy test
ACOS(config-policy)# forward-policy
```

4. Configure the action list `drop` to drop packets and log the event.

```
ACOS(config-policy-forward-policy)# action drop
ACOS(config-policy-forward-policy-action)# drop
ACOS(config-policy-forward-policy-action)# log
ACOS(config-policy-forward-policy-action)# exit
```

5. Configure the action list `int` to forward Internet traffic to the service group `gateway_http` and log the event. Service group `gateway_http` is a config example, IP or port in this service-group will not be used.

```
ACOS(config-policy-forward-policy)# action int
```

```
ACOS(config-policy-forward-policy-action)# forward-to-internet gateway_  
http  
ACOS(config-policy-forward-policy-action)# log  
ACOS(config-policy-forward-policy-action)# exit
```

6. Configure the source list `s1` to apply the `int` action for any requests that are not defined by a class-list or web-category-list.

```
ACOS(config-policy-forward-policy)# source s1  
ACOS(config-policy-forward-policy-source)# match-any  
ACOS(config-policy-forward-policy-source)# destination any action int  
ACOS(config-policy-forward-policy-source)# exit  
ACOS(config-policy-forward-policy)# exit  
ACOS(config-policy)# exit
```

7. Configure a wildcard SLB virtual server and its ports. To trigger SNI-based filtering, bind the template policy `test` to the port 443 HTTPS. The option `use-rcv-hop-for-resp` enables the ACOS device to send replies to clients back through the last hop on which the request for the virtual port's service was received.

```
ACOS(config)# slb virtual-server wildcard 0.0.0.0 acl 100  
ACOS(config-slb vserver)# port 0 others  
ACOS(config-slb vserver-vport)# service-group outside_tcp  
ACOS(config-slb vserver-vport)# use-rcv-hop-for-resp  
ACOS(config-slb vserver-vport)# no-dest-nat  
ACOS(config-slb vserver-vport)# exit  
  
ACOS(config-slb vserver)# port 0 tcp  
ACOS(config-slb vserver-vport)# service-group outside_tcp  
ACOS(config-slb vserver-vport)# use-rcv-hop-for-resp  
ACOS(config-slb vserver-vport)# no-dest-nat  
ACOS(config-slb vserver-vport)# exit  
  
ACOS(config-slb vserver)# port 0 udp  
ACOS(config-slb vserver-vport)# service-group outside_udp  
ACOS(config-slb vserver-vport)# use-rcv-hop-for-resp  
ACOS(config-slb vserver-vport)# no-dest-nat  
ACOS(config-slb vserver-vport)# exit  
ACOS(config-slb vserver)# port 443 https  
ACOS(config-slb vserver-vport)# service-group gateway_http
```

```
ACOS(config-slb vserver-vport)# source-nat pool snat2
ACOS(config-slb vserver-vport)# template policy test
ACOS(config-slb vserver-vport)# no-dest-nat port-translation
ACOS(config-slb vserver-vport)# exit
```

Consolidated Example

The following is the consolidated CLI example for SNI-based filtering:

```
slb server gateway 10.10.10.2
  port 80 tcp
!
slb service-group gateway_http tcp
  member gateway 80
!
slb template policy test
  forward-policy
    action drop
      drop
      log
    action int
      forward-to-internet gateway_http
      log
    source s1
      match-any
      destination any action int
!
slb virtual-server wildcard 0.0.0.0 acl 100
  port 0 others
    service-group outside_tcp
    use-rcv-hop-for-resp
    no-dest-nat
  port 0 tcp
    service-group outside_tcp
    use-rcv-hop-for-resp
    no-dest-nat
  port 0 udp
    service-group outside_udp
    use-rcv-hop-for-resp
    no-dest-nat
  port 443 https
```

```
source-nat pool snat2
service-group gateway_http
template policy test
no-dest-nat port-translation
```

GUI Configuration

The following procedure configures an ACOS device in transparent proxy mode to perform SNI-based filtering of traffic. Perform the following steps in order:

Step 1. Configuring the Real Server

1. Navigate to **ADC >> SLB >> Servers**.
2. Click **Create** and configure the following real server settings:
 - Name: **gateway**
 - Type: **IPv4**
 - Host: **10.10.10.2**
 - Action: **Enable**
3. Under **Port**, click **Create**, and configure the following port settings:
 - Port Number: **80**
 - Protocol: **TCP**
4. Click **Create**.

The real server is created.

Step 2. Configuring the Service Group

1. Navigate to **ADC >> SLB >> Service Groups**.
2. Click **Create** and configure the following settings:
 - Name: **gateway_http**
 - Protocol: **TCP**
3. Under **Member**, select **Create**.
The **Create Member** page is displayed.
4. Under **Choose Creation Type**, select **Existing Server**.
5. For **Server**, select **GW** from the drop-down menu.

6. For **Port**, select **80**.
7. Select **State** as **Enable**.
8. Click **Create**.

The real server **gateway** and port 80 are now associated with the service group **gateway_http**.

Step 3. Configuring the Policy Template

1. Navigate to **Security >> Forward Proxy >> Templates**.
2. Click **Create Policy**.
3. Specify the name as **Test**.
4. Under **Action Policies**, click **Add**.
The **Create Action Policy** page is displayed.
5. Specify the following to create an action called **drop**:
 - Name: **drop**
 - Action: **Drop**
 - Select **Logging**.
 - Select **OK**.
6. Under **Action Policies**, click **Add** again to create the action called **int**. Specify the following:
 - Name: **int**
 - Action: **forward-to-internet**
 - Service Group: **gateway_http**
 - Select **Logging**.
 - Select **OK**.
7. Actions **drop** and **int** are created under **Action Policies**.
8. Click **Source Policies**.
9. Click **Add**.
The **Add Source Policy** page is displayed.

10. Specify the following:
 - Name: **s1**
 - Match-Type: **Any**
11. Under Destination Rules, click **Add**.
12. Select Match Type as **Any** and Action as **int**.
13. Click **OK**.
14. Click **Add Template** to create the policy template.

Step 4. Configuring the Virtual Server and vPort

1. Navigate to **ADC >> SLB >> Virtual Servers**.
2. Click **Create**.
The **Create Virtual Server** page is displayed.
3. For **Name**, enter **wildcard**, and configure the following
4. Select **Wildcard**.
5. For Address Type, select **IPv4**.
6. For IP Address, select **0.0.0.0**.
7. For Action, select **Enable**.
8. For Source NAT Pool, select **SNAT2**.
You must have already created SNAT2.
Click **Create** at **ADC >> IP Source NAT >> IPv4 Pools**.
9. For Access List, select **100**.
10. Under **Virtual Port**, click **Create**.
The **Create Virtual Port** page is displayed.
11. Configure the following:
 - For Protocol, select **HTTPS**.
 - For Port, select **443**.
 - For Action, select **Enable**.
 - For Service Group, select **gateway_http**.

12. Expand **General Fields** and select the following:
 - No Dest NAT.
 - Port Translation.
13. Expand **Templates**.
14. For Template Policy, select **test**.
15. Click **Create**.
The Virtual Port is created and added to the virtual server.

URL Filtering Based on SAN

SAN is an extension which allows for multiple domain names to be protected with a single certificate. The SAN extension can include email addresses, IP addresses, URIs, DNS names, and other names.

ACOS supports configuring SAN-based URL filtering only when the SNI extension is not found in the client hello message. By default, the SAN filtering is disabled. Use the `no-sni-allow` command to allow requests to be forwarded in the absence of SNI, and then use the `ssli-url-filtering-san enable-san` command to configure SAN filtering on transparent proxy SSLi traffic. When a SAN certificate is received, the domain names in SAN and the common name of certificate subject or certificate issuer is used to match the class-list to enforce the policy. A client-SSL template is necessary for SAN URL filtering.

Configuring SAN-based Filtering by CLI

The following procedure configures an ACOS device in transparent proxy mode to perform SAN-based filtering of traffic.

1. Configure the real server called `gateway` at IP address `10.10.10.2`. Port `80` is added for TCP traffic.

```
ACOS(config)# slb server gateway 10.10.10.2
ACOS(config-real server)# port 80 tcp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
```

2. Configure a TCP service group called `gateway_http` and associate the real server

gateway and port 80 to the service group.

```
ACOS(config)# slb service-group gateway_http tcp
ACOS(config-slb svc group)# member gateway 80
ACOS(config-slb svc group-member:80)# exit
ACOS(config-slb svc group)# exit
```

3. Configure client-ssl template for SAN URL filtering. A server-ssl template is needed to be set on outside device, refer [Basic Transparent Proxy Configuration \(CLI Example\)](#).

```
ACOS(config)#slb template client-ssl c1
ACOS(config-client ssl)# forward-proxy-ca-certificate Cert123.pem key
key123
ACOS(config-client ssl)# forward-proxy-enable
```

4. Configure a template policy called `test_policy`. Enable `forward-policy` to specify permitted traffic destinations and sources along with the actions to apply.

```
ACOS(config)# slb template policy test_policy
ACOS(config-policy)# forward-policy
```

5. Configure the SAN-based SSLi URL filtering for HTTPS vport. By default, SAN filtering is disabled for intercepted URLs and enabled for bypassed URLs. Configure enable SAN filtering for intercepted URLs and disable SAN filtering for bypassed URLs.

```
ACOS(config-policy-forward-policy)# ssli-url-filtering no-sni-allow
ACOS(config-policy-forward-policy)# ssli-url-filtering-san enable-san
ACOS(config-policy-forward-policy)# ssli-url-filtering-san intercepted-
san-enable
ACOS(config-policy-forward-policy)# ssli-url-filtering-san bypassed-san-
disable
```

6. Configure the action list `drop` to drop packets and log the event.

```
ACOS(config-policy-forward-policy)# action drop
ACOS(config-policy-forward-policy-action)# drop
ACOS(config-policy-forward-policy-action)# log
ACOS(config-policy-forward-policy-action)# exit
```

7. Configure the action list `int` to forward Internet traffic to the service group

`gateway_http` and log the event.

```
ACOS(config-policy-forward-policy)# action int
ACOS(config-policy-forward-policy-action)# forward-to-internet gateway_
http
ACOS(config-policy-forward-policy-action)# log
ACOS(config-policy-forward-policy-action)# exit
```

8. Configure the source list `s1` to apply the `int` action for any requests that are not defined by a class-list or web-category-list.

```
ACOS(config-policy-forward-policy)# source s1
ACOS(config-policy-forward-policy-source)# match-any
ACOS(config-policy-forward-policy-source)# destination any action int
ACOS(config-policy-forward-policy-source)# exit
ACOS(config-policy-forward-policy)# exit
ACOS(config-policy)# exit
```

9. Configure a wildcard SLB virtual server and its ports. To trigger SAN-based filtering without SSLi, bind the service group `gateway_http` and the template policy `test` to the port 443 HTTPS. See the ACOS CLI highlighted in blue in the following sample.

The option `use-rcv-hop-for-resp` enables the ACOS device to send replies to clients back through the last hop on which the request for the virtual port's service was received.

```
ACOS(config)# slb virtual-server wildcard 0.0.0.0 acl 100
ACOS(config-slbg vserver)# port 0 others
ACOS(config-slbg vserver-vport)# service-group outside_tcp
ACOS(config-slbg vserver-vport)# use-rcv-hop-for-resp
ACOS(config-slbg vserver-vport)# no-dest-nat
ACOS(config-slbg vserver-vport)# exit

ACOS(config-slbg vserver)# port 0 tcp
ACOS(config-slbg vserver-vport)# service-group outside_tcp
ACOS(config-slbg vserver-vport)# use-rcv-hop-for-resp
ACOS(config-slbg vserver-vport)# no-dest-nat
ACOS(config-slbg vserver-vport)# exit

ACOS(config-slbg vserver)# port 0 udp
```

```

ACOS(config-slb vserver-vport)# service-group outside_udp
ACOS(config-slb vserver-vport)# use-rcv-hop-for-resp
ACOS(config-slb vserver-vport)# no-dest-nat
ACOS(config-slb vserver-vport)# exit
ACOS(config-slb vserver)# port 443 https
ACOS(config-slb vserver-vport)# template client-ssl c1
ACOS(config-slb vserver-vport)# service-group gateway_http
ACOS(config-slb vserver-vport)# source-nat pool snat2
ACOS(config-slb vserver-vport)# template policy test
ACOS(config-slb vserver-vport)# no-dest-nat port-translation
ACOS(config-slb vserver-vport)# exit

```

Consolidated Example

The following is the consolidated CLI example for SAN-based filtering:

```

slb server gateway 10.10.10.2
    port 80 tcp
!
slb service-group gateway_http tcp
    member gateway 80
!
slb template client-ssl c1
forward-proxy-ca-certificate Cert123.pem key key123
forward-proxy-enable
!
slb template policy test_policy
    forward-policy
ssli-url-filtering no-sni-allow
ssli-url-filtering-san enable-san
ssli-url-filtering-san intercepted-san-enable
ssli-url-filtering-san bypassed-san-disable
action drop
    drop
    log
    action int forward-to-internet gateway_http
    log
source s1
    match-any
    destination any action int
!

```

```
slb virtual-server wildcard 0.0.0.0 acl 100
  port 0 others
    service-group outside_tcp
    use-rcv-hop-for-resp
    no-dest-nat
  port 0 tcp
    service-group outside_tcp
    use-rcv-hop-for-resp
    no-dest-nat
  port 0 udp
    service-group outside_udp
    use-rcv-hop-for-resp
    no-dest-nat
  port 443 https
    source-nat pool snat2
    service-group gateway_http
    template client-ssl c1
    template policy test
    no-dest-nat port-translation
```

Configuring SAN-based Filtering by GUI

The following procedure configures an ACOS device in transparent proxy mode to perform SAN-based filtering of traffic. Perform the following steps in order:

Step 1. Configuring the Real Server

1. Navigate to **ADC >> SLB >> Servers**.
2. Click **Create** and configure the following real server settings:
 - Name: gateway
 - Type: IPv4
 - Host: 10.10.10.2
 - Action: Enable
3. Under Port, click **Create**, and configure the following port settings:
 - Port Number: 80
 - Protocol: TCP

4. Click **Create**.

The real server is created.

Step 2. Configuring the Service Group

1. Navigate to **ADC >>SLB >>Service Groups**.
2. Click **Create** and configure the following settings:
 - Name: gateway_http
 - Protocol: TCP
3. Under Member, select Create.
The Create Member page is displayed.
4. Under **Choose Creation Type**, select **Existing Server**.
 - For Server, select **GW** from the drop-down menu.
 - For Port, select 80.
5. Select **State as Enable**.
6. Click **Create**.

The real server gateway and port 80 are now associated with the service group gateway_http.

Step 3. Configuring the client-ssl template

1. Click **ADC >> Templates >> SSL**.
2. Click **Create >> Client SSL**.
3. Specify the name as "c1".
4. Input "rootca.crt" in Forward Proxy CA Cert field.
5. Input "rootca-dec.key" in Forward Proxy CA Private Key field.
6. Check **Forward Proxy Enable**.
7. Click **OK**.

Step 4. Configuring the Policy Template

1. Click **Security >> Forward Proxy >> Templates**.
2. Click **Create Policy**.

3. Specify the name as Test_Policy.
4. Under Action Policies, click **Add**.
The Create Action Policy page is displayed.
5. Specify the following to create an action called drop:
 - Name: drop
 - Action: Drop
 - Select Logging.
 - Select OK.
6. Under Action Policies, click **Add** again to create the action called int.
Specify the following:
 - Name: int
 - Action: forward-to-internet
 - Service Group: gateway_http
 - Select Logging.
 - Select OK.
7. Actions drop and int are created under Action Policies.
8. Click **Source Policies**.
9. Click **Add**.
The Add Source Policy page is displayed.
10. Specify the following:
 - Name: s1
 - Match-Type: Any
11. Under Destination Rules, click Add.
12. Select Match Type as Any and Action as int.
13. Click **OK**.

14. Specify the following to enable SAN-based filtering of traffic:
 - For SSLi URL Filtering, select No SNI Allow.
 - For SSLi URL Filtering SAN, select Enable SAN, Bypassed SAN Disable, and Intercepted SAN Enable.
15. Click Add Template to create the policy template.

Step 5. Configuring the Virtual Server and vPort

1. Click **Create**.
The Create Virtual Server page is displayed.
2. For Name, enter wildcard, and configure the following
3. Select **Wildcard**.
4. For Address Type, select **IPv4**.
5. For IP Address, select **0.0.0.0**.
6. For Action, select **Enable**.
7. For Source NAT Pool, select SNAT2. You must have already created SNAT2.
8. Click **Create** at **ADC >> IP Source NAT >> IPv4 Pools**.
9. For Access List, select 100.
10. Under Virtual Port, click Create.
The Create Virtual Port page is displayed.
11. Configure the following:
 - For Protocol, select HTTPS.
 - For Port, select 443.
 - For Action, select Enable.
 - For Service Group, select gateway_http.
 - For Template Client SSL, select c1.
12. Expand General Fields and select the following:
 - No Dest NAT.
 - Port Translation.
13. Expand Templates.

14. For Template Policy, select **test**.
15. Click **Create**.
The Virtual Port is created and added to the virtual server.

Resolve Unknown URLs

Typically, when a client sent a request to an unknown URL, ACOS matched the URL with the configured web-category list to identify the web-category for the URL. If the URL did not match any web-category, the request is processed by marking the URL as uncategorized. The unknown URL is resolved later and added to the relevant web-category.

You can enable “Wait for Web-category to be resolved” in the Forward Policy/Client SSL template and ensure the request is not processed until the web-category is resolved for the unknown URL. When “Wait for Web-category to be resolved” is enabled, ACOS intercepts the incoming request and determines whether the request needs to be allowed or denied after the URL category is identified.

The steps below explain how ACOS performs the web-category lookup when “Wait for Web-category to be resolved” is enabled.

- A client sends a request to an unknown URL.
- ACOS intercepts the request and checks if the URL belongs to any web category in the local database. If Yes, the request is allowed or denied as per the defined policy.
 - If the URL does not belong to any web category in the local database, ACOS performs the web category lookup in the cloud cache.
 - If the URL belongs to any web category in the cloud cache, the request is allowed or denied as per the defined policy.
 - If the URL does not belong to any web category in the local database and cloud cache (i.e., uncategorized). ACOS submits the URL for categorization and waits for the web category to be resolved.
- After the category is resolved, ACOS updates the category list and processes the request as per the defined policy.

CLI Configuration

Policy Template:

```
ACOS(config)#slb template policy Policy-Template
ACOS(config-policy)#forward-policy
ACOS(config-policy-forward-policy)#require-web-category
```

Client SSL Template:

```
ACOS(config)#slb template client-ssl Client-SSL
ACOS(config-client ssl)#forward-proxy-bypass require-web-category
```

GUI Configuration

“Wait For Web Category” can be enabled in the following policies on the ACOS web interface:

Forward Proxy Policy Template

1. Navigate to **Security >> Forward Proxy >> Templates** tab.
2. Select Policy from the Create drop-down list.
3. On the Add Policy Template page, under the Action Policies section:
4. Select the Wait For Web Category check box.
5. Specify values for other parameters as required.
6. Click **Add Template**.

Forward Proxy Client SSL Template

1. Navigate to **Security >> Forward Proxy >> Templates** tab.
2. Select Client SSL from the Create drop-down list.
3. On the Create Client SSL Template page, under the Policies section:
4. Select the **Wait For Web Category** check box.
5. Specify values for other parameters as required.
6. Click **Add**.

SSLi Client SSL Template

1. Navigate to **Security >> SSLi >> Templates** tab.
2. Select Client SSL from the Create drop-down list.
3. On the Create Client SSL Template page, under the Policies section:
4. Select the **Wait For Web Category** check box.
5. Specify values for other parameters as required.
6. Click **Add**.

Customizable Response Codes on Blocking HTTP Requests

The standard status codes are configurable for blocking HTTP and HTTPS traffic for explicit proxy and for blocking HTTP traffic for transparent proxy. The feature is supported only with the drop-message option for forward policy in an SLB template. The configurable response code can be any integer between 100 to 599. The SLB template must then be associated with a virtual server.

The following is a list of the behavior for explicit proxy deployments.

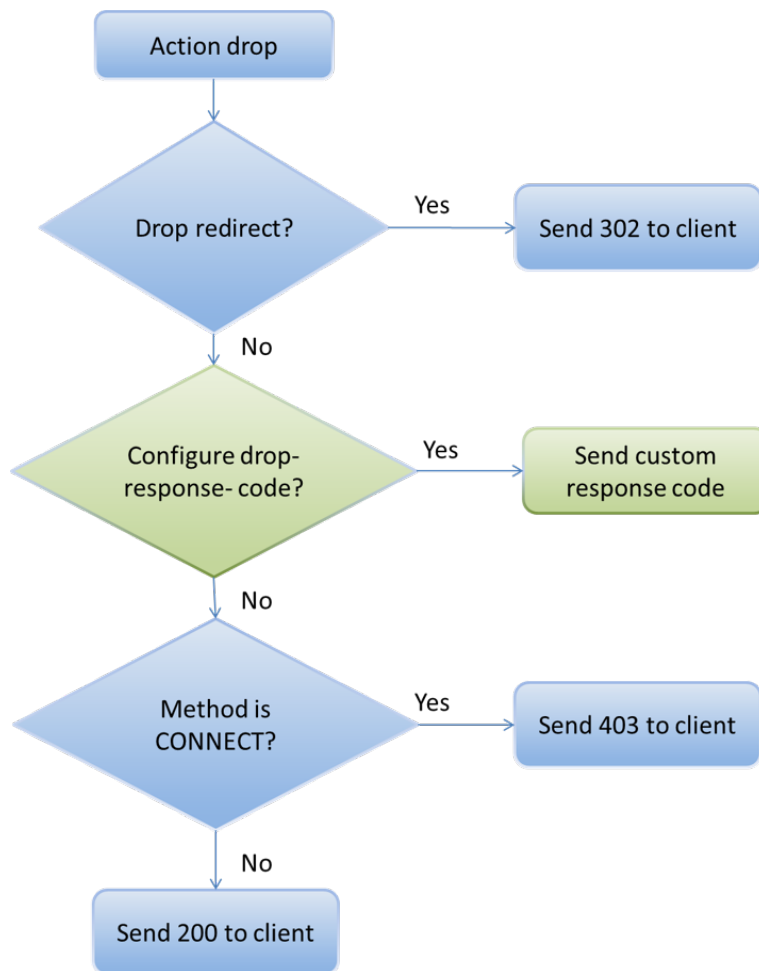
- For blocking HTTP requests-- Configurable (default 200)
- For blocking HTTPS requests-- Configurable (default 403) *HTTP header check on CONNECT method

The following is a list of the behavior for transparent proxy deployments.

- For blocking HTTP requests-- Configurable (default 200)
- For blocking HTTPS requests-- TCP RST *SNI header check

The following is the workflow of the behavior. Blue blocks depict original behavior. Green blocks depict the newly added feature.

Figure 83 : Workflow for the drop action



CLI Configuration

Create a policy template called `TEST`. Configure a drop code of 300. Perform the following steps:

1. Create the SLB template called `TEST`.

```
ACOS(config)#slb template policy TEST
```

2. Enable forward policy.

```
ACOS(config-policy)# forward-policy
```

3. Define the action for drop as `drop-response-code`.

```
ACOS(config-policy-forward-policy)# action drop
ACOS(config-policy-forward-policy-action)# drop-response-code ?
<100 - 599> Specify response code for drop action
ACOS(config-policy-forward-policy-action)# drop-response-code 300
```

4. Save the configuration.

GUI Configuration

Create a policy template called `TEST`. Configure a drop code of 300. Perform the following steps:

1. Navigate to **Security >> Forward Proxy >> Templates >> Create >> Create Policy**.
2. The **Add Policy** window is displayed.
3. Fill in the field for **Name** - Name of the policy, such as **TEST**.
4. Under **Action Policies** tab, click **Add**.
5. The **Create Action Policy** window is displayed.
6. Fill in the following details:
 - **Name** - Name of the action, such as `Drop_code`.
 - **Action** - Select one of the following:
 - Drop
 - Forward Request to Internet
 - Forward Request to Service Group
 - Forward Request to Proxy
7. For Drop, the following are the additional details:
 - **Drop message** -- Enter a customized drop message.
 - **Drop Response Code** -- Enter a code between 100 to 599, such as 300.
8. Click **OK**.
9. Click **Add Template** to add the policy.

Policy template was created successfully.

Logging HTTP Proxy

An HTTP log message can be displayed in the standardized Syslog format or Common Event Format (CEF). By default, the message is displayed in Syslog format. HTTP requests handled by this feature can be logged based on the outcome of the request:

- Permitted
- Denied (dropped)
- DNS failure or SNAT failure

Some examples of log messages in the Syslog and CEF format for the explicit HTTP-proxy feature are explained in the next sections. Each of them shows information about an HTTP request intercepted by the HTTP virtual port.

Syslog

The following requests are from source (client) 10.10.10.15. The client IP address is translated to NAT address 172.31.31.0. ACOS replaces the source IP address of requests with this NAT address before forwarding them to the destinations. There are three cases shown below for drop, forward to service group, and forward to Internet.

Drop:

```
Apr 01 2015 14:16:47 Info      [ACOS]:Proxy GET [drop- (s1
priority#1)]:www.a4.com url http://10.10.10.15 from 10.10.10.10:14993, to
0.0.0.0:0
```

Forward to service-group:

```
Apr 01 2015 14:16:44 Info      [ACOS]:Proxy GET [server- (s1
priority#500)]:www.a3.com url http://10.10.10.15 from 10.10.10.10:65518,
snat 172.31.31.99:2194 to 172.31.31.20:80 out-bytes: 196
```

```
Apr 01 2015 14:16:44 Info      [ACOS]:Proxy GET [select-server- (s1
priority#500)]:www.a3.com url http://10.10.10.15 from 10.10.10.10:65518,
to 0.0.0.0:0 out-bytes: 184
```

Forward to internet:

```
Apr 01 2015 14:16:40 Info      [ACOS]:Proxy GET [internet- (s1
priority#1024)]:www.a1.com url http://10.10.10.15 from 10.10.10.10:59605,
snat 172.31.31.99:2185 to 172.31.31.22:80 out-bytes: 196
```

In the case when a web category is found, it will also appear in the log message.

```
Jun 18 2020 15:44:00 Info      [ACOS]:Proxy [response-code 200] client
10.1.1.1:41191, server 172.16.1.1:3128, in-bytes=1367
Jun 18 2020 15:44:00 Info      [ACOS]:Proxy GET[proxy- (s1 priority#1
category auctions reputation trustworthy(88))]:www.ebay.com url
http://www.ebay.com/mng/1000.txt client 10.1.1.1:41191, snat
172.1.99.1:2075 server 172.16.1.1:3128 out-bytes:229
Jun 18 2020 15:44:00 Info      [ACOS]:Proxy GET[select-proxy- (s1
priority#1 category auctions reputation trustworthy(88))]:www.ebay.com url
http://www.ebay.com/mng/1000.txt client 10.1.1.1:41191, server 0.0.0.0:0
out-bytes:229
```

Log Message Format

Log messages for the explicit HTTP-proxy feature show the following fields:

```
timestamp severity [module]:feature method [action -(filter)]:host url
url_text from source_ip:source_port, snat snat_ip:snat_port to des-
tination_ip:destination_port out-bytes or in-bytes
```

[Table 18](#) describes the fields from the Syslog Log Message Format.

Table 18 : Log Message Field Descriptions for Syslog

Field	Description
timestamp	System time on the ACOS Device when the message was generated.
severity	Message severity level.
module/feature	System module and feature.
method	HTTP method, typically get, post, connect.
action	Action performed on the request: Internet (forward to Internet), or drop.
filter	Source rule name, priority (rule) number within that group that matched the request. Category web-category listed if applicable. A “-1” is returned if nothing is found.
host	Destination host or domain name of the request.
url url_text	URL of the request.
from source_ip:source_	Source IP address and protocol port of the request.

Table 18 : Log Message Field Descriptions for Syslog

Field	Description
port	
snat snat_ip:snat_port	If source NAT was provided, the NAT IP address and pool that ACOS assigned to the source before forwarding the request. (If no source NAT was provided, this field will not be displayed.)
to destination_ip:destination_port	Destination IP address and protocol port of the request.
out-bytes or in-bytes	Number of bytes of the first packet transferred outbound or inbound.

CEF

The following are some examples to show the log message format in CEF.

This example shows the log message format when a HTTP request is forwarded to Internet. The request is received from the source (client) 172.16.1.181. ACOS replaces the source IP address of the request with 192.168.92.111 before forwarding it to the destination.

```
CEF:0|A10|ADC|4.1.4-FR8403|Fwd Policy 545290705757536257|Forward
Request|6|src=172.16.1.181 spt=36676 dst=192.168.90.160 dpt=80
sourceTranslatedAddr=192.168.92.111 sourceTranslatedPort=2056
requestMethod=GET request=http://192.168.90.160/ out=196 suser=user1 act=
[internet- (ANY priority#0)]
```

[Table 19](#) describes the fields from the CEF Log Message Format.

Table 19 : Log Message Field Descriptions for CEF - Forward Request

Field	Description
src	Source IP address of the request.
spt	Source port of the request.
dst	Destination IP address of the request.
dpt	Destination port of the request.
sourceTranslatedAddr	IP address of the translated source.

Table 19 : Log Message Field Descriptions for CEF - Forward Request

Field	Description
sourceTranslatedPort	Port after it was translated.
requestMethod	Method used to access a URL.
request	URL accessed during the HTTP request.
out	Number of bytes of the first packet transferred outbound.
suser	Name of the source user.
act	Action performed on the request — Internet or drop.

This example shows the log message format when a specific HTTP request is successfully received.

```
CEF:0|A10|ADC|4.1.4-FR8403|Fwd Policy 545290705757536258|Receive
Response|6|src=172.16.1.181 spt=36676 dst=192.168.90.160 dpt=80 response-
code=200 in=268
```

[Table 20](#) describes the fields from the CEF Log Message Format.

Table 20 : Log Message Field Descriptions for CEF - Receive Response

Field	Description
src	Source IP address of the request.
spt	Source port of the request.
dst	Destination IP address of the request.
dpt	Destination port of the request.
response-code	Response status code of the HTTP request.
in	Number of bytes of the first packet transferred inbound.

Displaying HTTP Statistics

Displaying HTTP Explicit Proxy Statistics (CLI Example)

To display statistics for HTTP explicit proxy, use the **show slb http-proxy** command. The following shows a sample output for **show slb http-proxy**.

```
ACOS(config)# show slb http-proxy
```

```
Total
```

```
-----  
Curr Proxy Conns 0  
Total Proxy Conns 7  
HTTP requests 7  
HTTP requests(succ) 7  
HTTP requests(CONNECT) 0  
HTTP requests enter SSLi 0  
HTTP req (cache succ) 0  
No proxy error 0  
Client RST 0  
Server RST 0  
No tuple error 0  
Parse req fail 0  
Server selection fail 0  
Fwd req fail 0  
Fwd req data fail 0  
Req retransmit 0  
Req pkt out-of-order 0  
Server reselection 0  
Server premature close 0  
Server conn made 7  
Source NAT failure 0  
Tot data before compress 0  
Tot data after compress 0  
Request over limit 0  
Request rate over limit 0  
  
Close on DDoS 0
```

Displaying HTTP Explicit Proxy Statistics (GUI Example)

To display statistics for HTTP explicit proxy, take the following steps:

1. Navigate to **Security>> Forward Proxy**.
2. Click the **Statistics** tab.
3. Click the **HTTP Proxy** tab, if not already selected.

Displaying Statistics for Forward Policy

Statistics for Forward Policy can be displayed using the CLI or GUI.

CLI Configuration

To display forward policy statistics, use the `show slb template policy forward-policy-stats` command:

The statistics display the following fields:

- Requests dropped
- Source NAT failure
- Unresolved DNS requests
- Outstanding DNS requests

The following shows a sample output for `show slb template policy forward-policy-stats`.

```
ACOS(config)# show slb template policy forward-policy-stats

slb template policy name: explicit-proxy-fwd-policy
Source NAT failure:      0
Unresolved DNS requests: 329
Outstanding DNS requests: 369
```

GUI Configuration

To display the statistics for the forward policy from the GUI, take the following steps:

1. Navigate to **Security >> Forward Proxy >> Statistics >> Policy** tab.
2. In the Policy Template field, use the search function to locate your forward policy template.
3. The statistics for your policy template is displayed.
4. Click **Refresh** to update or **Clear** to clear out the existing statistics.

Displaying or Clearing Learned Cache Entries

To display learned DNS cache entries, use the **show dns-cache** command. The following is a sample output.

```
ACOS(config)# show dns-cache
```

Record Name	A/CNAME Record	DNS
Server	TTL	
www.a10networks.com		
50.112.95.73	192.168.1.50	15
ajax.googleapis.com	216.58.192.42	192.168.1.50
125		
netapp.com	137.135.69.133	192.168.1.50
280		
www.ietf.org		
104.20.0.85	192.168.1.50	298
www.netapp.com		
172.231.39.203	192.168.1.50	1
api.demandbase.com	54.201.118.54	192.168.1.50
45		
www.google.com		
74.125.239.113	192.168.1.50	207
cloud.typography.com		
23.205.66.110	192.168.1.50	5
ietf.org	4.31.198.44	192.168.1.50
1787		
assets.adobedtm.com		
23.205.61.17	192.168.1.50	5

To clear learned cache entries, use the **clear dns-cache** command.

Displaying HTTP Explicit Proxy Category-Lists Counters

Counters for HTTP Web Category Lists can be displayed using the CLI or GUI.

Displaying HTTP Explicit Proxy Web-Category Category-Lists Counters (CLI Example)

To display HTTP Explicit Proxy web-category counters, use the command **show counters web-category category-list**. The number of hits is displayed for all the various categories.

The following is a partial sample output using category-list search that contains the social network category configured in [CLI Configuration](#):

```
ACOS(config)# show counters web-category category-list Search-Sites
uncategorized category hits                                0
real estate category hits                                  0
computer and internet security category hits              0
financial services category hits                          0
business and economy category hits                        0
computer and internet info category hits                  0
auctions category hits                                    0
shopping category hits                                    0
cult and occult category hits                              0
travel category hits                                      0
drugs category hits                                       0
adult and pornography category hits                       0
home and garden category hits                             0
military category hits                                    0
social network category hits                              3
dead sites category hits                                  0
stock advice and tools category hits                      0
training and tools category hits                          0
dating category hits                                      0
sex education category hits                               0
religion category hits                                    0
entertainment and arts category hits                      0
...
```

Displaying HTTP Explicit Proxy Web-Category Category-Lists Counters (GUI Example)

Take the following steps:

1. Navigate to Security>> Web Categories.
2. Click the Statistics tab.

3. Click Category Hits tab if not already there.
4. Use the search function to locate your category list.
5. Click Refresh, and all the web categories from your category list is displayed with the number of hits.

Proxy Chaining SLB

Proxy chaining involves the use of two or more proxy servers, where client requests are forwarded from one proxy to another before reaching the destination server. When an ACOS device is deployed as an explicit proxy, HTTP requests are normally processed and modified by the proxy.

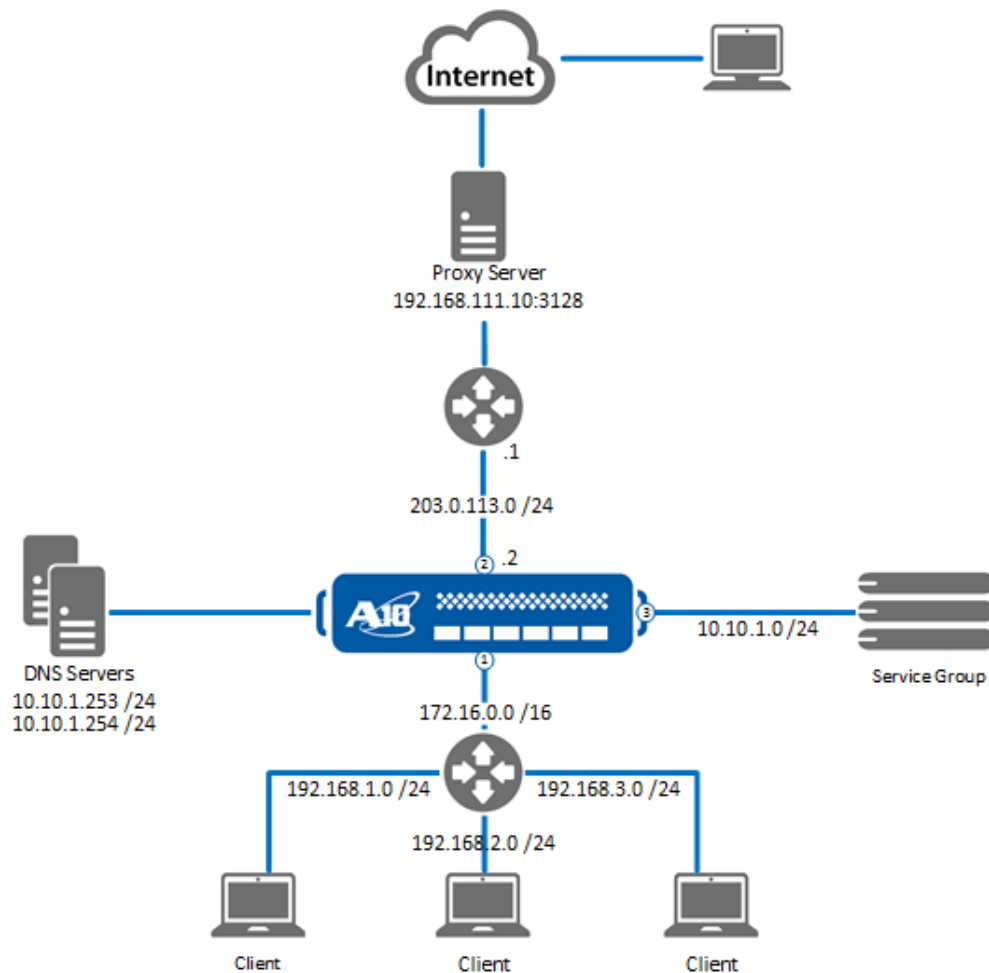
For proxy chaining scenarios, modifying the HTTP header can cause compatibility issues with upstream explicit proxy servers. This potentially results in dropped requests. To address this, ACOS supports proxy chaining through the use of the **forward-to-proxy** action in an SLB policy template. This action ensures that the original HTTP header is preserved when forwarding requests to an upstream proxy.

ACOS requires the following configuration for proxy chaining:

- An SLB server and service group representing the upstream proxy
- A policy template with a forward policy that explicitly forwards traffic to the upstream proxy

For more information on explicit proxy in an SSLi configuration, refer *SSLi Configuration Guide*.

Figure 84 : Explicit HTTP Proxy Configuration with Upstream Proxy Server (Proxy Chaining)



Configuring Proxy Chaining

ACOS supports the following proxy chaining:

- Explicit Proxy Chaining
- HTTP/2 Explicit Proxy Chaining

Explicit Proxy Chaining

In explicit proxy chaining, the ACOS device operates as a downstream explicit proxy and forwards client requests to an upstream explicit proxy. The forwarding behavior

is explicitly defined using the `forward-to-proxy` command, which prevents HTTP header modification and allows interoperability with the upstream proxy.

CLI Configuration

The following configuration steps set up a proxy-chaining environment, using the [Basic Explicit Proxy Configuration](#) and [Advanced Explicit Proxy Configuration](#) as an initial template.

1. Define the upstream proxy server and service group. Here, the upstream `proxy-server` has been configured with the IP address of 192.168.111.10 and port 3128.

```
ACOS(config)# slb server proxy-server 192.168.111.10
ACOS(config-real server)# port 3128 tcp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
ACOS(config)# slb service-group psg-3128 tcp
ACOS(config-slb svc group)# member proxy-server 3128
ACOS(config-slb svc group-member:3128)# exit
ACOS(config-slb svc group)# exit
ACOS(config)# ip nat pool Internet_Pool 203.0.113.5 203.0.113.5 netmask
/32
```

2. Modify the existing forward policy to redirect Internet-bound traffic to the upstream proxy by using the `forward-to-proxy` command with service group `psg-3128`. This replaces the `forward-to-internet` configuration used in the Basic Explicit Proxy setup, while retaining source NAT and logging.

```
ACOS(config)# slb template policy Explicit_Proxy
ACOS(config-policy)# forward-policy
ACOS(config-policy-forward-policy)# action Permit_to_Internet
ACOS(config-policy-forward-policy-action)# forward-to-proxy psg-3128
snat Internet_Pool
ACOS(config-policy-forward-policy-action)# log
ACOS(config-policy-forward-policy-action)# exit
ACOS(config-policy-forward-policy)# exit
ACOS(config-policy)# exit
```


HTTP/2 Explicit Proxy Chaining

HTTP/2 explicit proxy chaining extends explicit proxy chaining by enabling end-to-end HTTP/2 support across the proxy chain. In this proxy chaining, HTTP/2 is preserved between the client, downstream proxy, and upstream proxy. Thus, it allows HTTP/2 multiplexing and improves performance.

To enable HTTP/2 proxy chaining:

1. The `forward-proxy-end-to-end-http2-support` option must be configured on the SLB policy template.
2. The `support-http2` option must be configured on:
 - The SLB server port (upstream proxy connection)
 - The SLB virtual-server port (client-facing) the `support-http2` option must be configured on

CLI Configuration

The following configuration steps set up an HTTP/2 proxy-chaining environment. The downstream proxy forwards traffic to an upstream explicit proxy at 192.168.111.10:3128 using HTTP/2.

1. Create an SLB server entry for the upstream explicit proxy and enable HTTP/2 on the server port.

```
ACOS(config)# slb server proxy-server 192.168.111.10
ACOS(config-real server)# port 3128 tcp
ACOS(config-real server-node port)# support-http2
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
```

2. Create a service group and add the upstream proxy server as a member.

```
ACOS(config)# slb service-group psg-3128 tcp
ACOS(config-slb svc group)# member proxy-server 3128
ACOS(config-slb svc group-member:3128)# exit
ACOS(config-slb svc group)# exit
```

3. Create a source NAT pool for Internet-bound traffic forwarded to the upstream proxy.

```
ACOS(config)# ip nat pool Internet_Pool 203.0.113.5 203.0.113.5 netmask
/32
```

4. Modify the SLB policy template to enable end-to-end HTTP/2 support for forward-proxy traffic and forward requests to the upstream explicit proxy using the service group.

```
ACOS(config)# slb template policy Explicit_Proxy
ACOS(config-policy)# forward-proxy-end-to-end-http2-support
ACOS(config-policy)# forward-policy
ACOS(config-policy-forward-policy)# action Permit_to_Internet
ACOS(config-policy-forward-policy-action)# forward-to-proxy psg-3128
snat Internet_Pool
ACOS(config-policy-forward-policy-action)# log
ACOS(config-policy-forward-policy-action)# exit
ACOS(config-policy-forward-policy)# exit
ACOS(config-policy)# exit
```

5. Enable HTTP/2 on the explicit proxy virtual server port.

```
ACOS(config)# slb virtual-server vs1 10.10.10.10
ACOS(config-slb vserver)# port 443 http
ACOS(config-slb vserver-vport)# support-http2
ACOS(config-slb vserver-vport)# exit
ACOS(config-slb vserver)# exit
```

GUI Configuration

The upstream proxy-server is configured with IP address of 192.168.111.10 and port 3128. Edit the existing Explicit_Proxy policy template with the proxy server information and service group. by taking these steps:

1. Navigate to **Security>> Forward Proxy >> Templates** tab.
2. Click **Edit** for Explicit_Proxy policy.
3. Click **Edit** for Permit_to_Internet action policy.
4. Click the **Action** drop-down menu, and click **Forward Request to Proxy**.
5. In the Service Group field, click +.
6. In the Name field, enter "psg-3128".

7. Click **+ Add Member**.
8. In the Name field, click **+** to add a server.
9. In the Name field, enter “proxy-server”.
10. In the Host field, enter “192.168.111.0”.
11. Click **+ Add Port**.
12. In the Port field, enter “3128”. click **Apply**, then click **OK**.
13. In the Port field for Service Group, enter “3128”. click **Apply**, then click **OK**.
14. In the IPv4 Pool field, select Internet_Pool.
15. Click the **Logging** check box to enable logging.
16. Click **Update**.

Explicit Proxy with AAM Policy

An explicit proxy can be configured to use aam authorization policy to determine user and group membership. Configuration is done at the source level of a forward-policy, where a source rule can have an aam authorization policy bound to it through the `match-authorize-policy` command. Use the `priority` command to determine the order of which configured match-authorize-policy should be checked within an slb template policy. For more information about the CLI commands, refer to “forward-policy” in the SLB Policy Template section of the *Command Line Interface Reference*.

For explicit proxy client authentication using form-based-logon in an SSLi set up in conjunction with this feature, to ensure the explicit proxy client authentication occurs, configure “match-any” or “match-class-list” in the source rule so it matches the first request in the slb policy template.

- Only LDAP and RADIUS servers can be used to obtain membership information.
- Authorization policies can be configured at a global level under an aaa-policy or at a source level under a forward-policy.
- Only one server or service group is allowed per template policy to obtain membership information.

- Server or service-group configurations in an authorization policy template linked to a forward-policy template at a source level are ignored.
- Form-based and SAML authentication is supported.

CLI Configuration

AAM must initially be configured.

If configuring AAM with an explicit proxy or AAM with an explicit proxy + SSLi configuration, the authentication logon methods supported are: HTTP basic, Kerberos and Form Based.

For more information about AAM configuration, see the Application Access Management Guide.

The following is an example of Windows Integrated Authentication (WIA), using Kerberos for authentication and Lightweight Directory Access Protocol (LDAP) for authorization.

NOTE: This example assumes a PC has been set up as a WIA user-agent.

1. Configure a NAT pool, EP_NAT, for use with explicit proxy.

```
ACOS(config)# ip nat pool EP_NAT 192.168.92.111 192.168.92.111 netmask /32
```

2. To set up an explicit proxy, a service group is needed as a stub. So a dummy server and service-group are created, EP_DUMMY, in addition to the real server and service group.

```
ACOS(config)# slb server EP_DUMMY 10.0.0.1
ACOS(config-real server)# health-check-disable
ACOS(config-real server)# port 80 tcp
ACOS(config-real server-node port)# health-check-disable
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
ACOS(config)# slb service-group EP_DUMMY tcp
ACOS(config-slb svc group)# health-check-disable
ACOS(config-slb svc group)# member EP_DUMMY 80
ACOS(config-slb svc group-member:80)# exit
```

```

ACOS(config-slb svc group)# exit
ACOS(config)# slb server HTTP_PROXY 192.168.98.199
ACOS(config-real server)# port 80 tcp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
ACOS(config)# slb service-group PROXY_GROUP tcp
ACOS(config-slb svc group)# member HTTP_PROXY 80
ACOS(config-slb svc group-member:80)# exit
ACOS(config-slb svc group)# exit

```

3. Configure a basic forward policy, using the stub EP_DUMMY and the NAT pool are configured.

```

ACOS(config)# slb template policy EP_FORWARD_1
ACOS(config-policy)# forward-policy
ACOS(config-policy-forward-policy)# action TO_INET
ACOS(config-policy-forward-policy-action)# forward-to-internet EP_DUMMY
snat EP_NAT
ACOS(config-policy-forward-policy-action)# log
ACOS(config-policy-forward-policy-action)# exit
ACOS(config-policy-forward-policy)# source ANY
ACOS(config-policy-forward-policy-source)# match-any
ACOS(config-policy-forward-policy-source)# destination any action TO_
INET
ACOS(config-policy-forward-policy-source)# exit
ACOS(config-policy-forward-policy)# exit
ACOS(config-policy)# exit

```

4. Set up the explicit proxy, using the service-group and our policy.

```

ACOS(config)# slb virtual-server EP 192.168.91.110
ACOS(config-slb vserver)# port 8081 http
ACOS(config-slb vserver-vport)# service-group EP_DUMMY
ACOS(config-slb vserver-vport)# template policy EP_FORWARD_1
ACOS(config-slb vserver-vport)# exit
ACOS(config-slb vserver)# exit

```

5. AAM is set up for Kerberos authentication, by configuring an aam authentication server along with the appropriate supporting aam templates and policies.

- For WIA Kerberos, the logon type must use http-authenticate, and auth-method must be configured as negotiate.

```
ACOS(config)# aam authentication logon http-authenticate NEGO
ACOS(config-http-authenticate auth logon:...)# auth-method negotiate
enable
ACOS(config-http-authenticate auth logon:...)# exit
```

- For WIA Kerberos, the user-agent that will access our vport must be a member of the realm.

```
ACOS(config)# aam authentication server windows KDC
ACOS(config-windows auth server:KDC)# host 192.168.221.50
ACOS(config-windows auth server:KDC)# realm EXAMPLEREALM.COM
ACOS(config-windows auth server:KDC)# exit
```

- For WIA Kerberos, a service principal name (SPN), is needed for the key distribution center (KDC), and requires the following: the KDC realm, account of the SPN owner, and the SPN name. So, if we define the SPN as axproxy.exemplerealm.com, the user-agent must use axproxy.exemplerealm.com as the name of the explicit proxy.

```
ACOS(config)# aam authentication account kerberos-spn SPN
ACOS(config-kerberos-spn:SPN)# realm EXAMPLEREALM.COM
ACOS(config-kerberos-spn:SPN)# account Owner
ACOS(config-kerberos-spn:SPN)# service-principal-name
HTTP/axproxy.exemplerealm.com
ACOS(config-kerberos-spn:SPN)# password PASSWORD
ACOS(config-kerberos-spn:SPN)# exit
```

6. Configure an aam authentication template for WIA Kerberos. An additional template is configured specifically for tracking session use, using auth-sess-mode.

```
ACOS(config)# aam authentication template NEGO_KRB_C
ACOS(config-auth template:NEGO_KRB_C)# logon NEGO
ACOS(config-auth template:NEGO_KRB_C)# server KDC
ACOS(config-auth template:NEGO_KRB_C)# account SPN
ACOS(config-auth template:NEGO_KRB_C)# exit
ACOS(config)# aam authentication template NEGO_KRB_I
ACOS(config-auth template:NEGO_KRB_I)# auth-sess-mode ip-based
ACOS(config-auth template:NEGO_KRB_I)# logon NEGO
```

```
ACOS(config-auth template:NEGO_KRB_I) # server KDC
ACOS(config-auth template:NEGO_KRB_I) # account SPN
ACOS(config-auth template:NEGO_KRB_I) # exit
```

7. Configure LDAP, starting with configuring aam authentication server.

```
ACOS(config) # aam authentication server ldap AD_LDAP
ACOS(config-ldap auth server:AD_LDAP) # host 192.168.230.101
ACOS(config-ldap auth server:AD_LDAP) # base
cn=Users,dc=examplerealm,dc=com
ACOS(config-ldap auth server:AD_LDAP) # admin-dn
cn=Administrator,cn=Users,dc=examplerealm,dc=com
ACOS(config-ldap auth server:AD_LDAP) # admin-secret PASSWORD
ACOS(config-ldap auth server:AD_LDAP) # exit
```

8. Create our authorization server configuration that is added to the aam aaa-policy **NEGO_KRB_authen** (step 11) with aam authorization policy **SERVER_PROVIDER**. To set the aam authorization policy as a server provider, the **forward-policy-authorize-only** configuration is added.

```
ACOS(config) # aam authorization policy SERVER_PROVIDER
ACOS(config-authorization policy:SERVER_P...) # server AD_LDAP
ACOS(config-authorization policy:SERVER_P...) # forward-policy-
authorize-only
ACOS(config-authorization policy:SERVER_P...) # exit
```

9. Bind the aam-template aaa-policy and configured to provide the authorization server, **SERVER_PROVIDER**, for our forward-policy through the **authorize-policy** command. This policy is added to the explicit proxy.

```
ACOS(config) # aam aaa-policy NEGO_KRB_authen
ACOS(config-aaa policy:NEGO_KRB_authen) # aaa-rule 10
ACOS(config-aaa policy:NEGO_KRB_authen-aa...) # authentication-template
NEGO_KRB_I
ACOS(config-aaa policy:NEGO_KRB_authen-aa...) # authorize-policy SERVER_
PROVIDER
ACOS(config-aaa policy:NEGO_KRB_authen-aa...) # exit
ACOS(config-aaa policy:NEGO_KRB_authen) # exit
ACOS(config) # slb virtual-server EP 192.168.91.110
ACOS(config-sl b vserver) # port 8082 http
ACOS(config-sl b vserver-vport) # service-group EP_DUMMY
```

```
ACOS(config-slb vserver-vport)# template policy EP_FORWARD_1
ACOS(config-slb vserver-vport)# aaa-policy NEGO_KRB_authen
ACOS(config-slb vserver-vport)# exit
ACOS(config-slb vserver)# exit
```

10. Configure two authorization policies, each with their own attribute values for use with AAM check for the explicit proxy's forward policy.

```
ACOS(config)# aam authorization policy GROUP_INET
ACOS(config-authorization policy:GROUP_I...)# attribute-rule 1
ACOS(config-authorization policy:GROUP_I...)# server AD_LDAP
ACOS(config-authorization policy:GROUP_I...)# attribute 1 memberOf
attr-type string sub-string CN=INET
ACOS(config-authorization policy:GROUP_I...)# exit
ACOS(config)# aam authorization policy GROUP_PROXY_NEED
ACOS(config-authorization policy:GROUP_PR...)# attribute-rule 1
ACOS(config-authorization policy:GROUP_PR...)# server AD_LDAP
ACOS(config-authorization policy:GROUP_PR...)# attribute 1 memberOf
attr-type string sub-string CN=PROXY_NEED
ACOS(config-authorization policy:GROUP_PR...)# exit
```

Now, the forward policy is configured. The default source drops all requests, while the defined source GROUP_INET and GROUP_PROXY are routed through the use of the **match-authorize-policy**. The command **priority** is used to determine the order for authorization checking and cannot be the same value, unless it is the default value of zero.

```
ACOS(config)# slb template policy EP_FORWARD_3
ACOS(config-policy)# forward-policy
ACOS(config-policy-forward-policy)# action DROP
ACOS(config-policy-forward-policy-action)# drop
ACOS(config-policy-forward-policy-action)# log
ACOS(config-policy-forward-policy-action)# exit
ACOS(config-policy-forward-policy)# action TO_INET
ACOS(config-policy-forward-policy-action)# forward-to-internet EP_DUMMY
snat EP_NAT
ACOS(config-policy-forward-policy-action)# log
ACOS(config-policy-forward-policy-action)# exit
ACOS(config-policy-forward-policy)# action TO_PROXY
ACOS(config-policy-forward-policy-action)# forward-to-proxy PROXY_GROUP
```



```

ACOS(config-policy-forward-policy-action)# log
ACOS(config-policy-forward-policy-action)# exit
ACOS(config-policy-forward-policy)# source ANY
ACOS(config-policy-forward-policy-source)# match-any
ACOS(config-policy-forward-policy-source)# destination any action DROP
ACOS(config-policy-forward-policy-source)# exit
ACOS(config-policy-forward-policy)# source GROUP_INET
ACOS(config-policy-forward-policy-source)# match-authorize-policy
GROUP_INET
ACOS(config-policy-forward-policy-source)# priority 1000
ACOS(config-policy-forward-policy-source)# destination any action TO_
INET
ACOS(config-policy-forward-policy-source)# exit
ACOS(config-policy-forward-policy)# source GROUP_PROXY_NEED
ACOS(config-policy-forward-policy-source)# match-authorize-policy
GROUP_PROXY_NEED
ACOS(config-policy-forward-policy-source)# priority 999
ACOS(config-policy-forward-policy-source)# destination any action TO_
PROXY
ACOS(config-policy-forward-policy-source)# exit
ACOS(config-policy-forward-policy)# exit
ACOS(config-policy)# exit

```

11. Bind the policy to the explicit proxy.

```

ACOS(config)# slb virtual-server EP 192.168.91.110
ACOS(config-slb vserver)# port 8083 http
ACOS(config-slb vserver-vport)# service-group EP_DUMMY
ACOS(config-slb vserver-vport)# template policy EP_FORWARD_3
ACOS(config-slb vserver-vport)# aaa-policy NEGO_KRB_authen
ACOS(config-slb vserver-vport)# exit
ACOS(config-slb vserver)# exit

```

12. LDAP authorization needs to be configured through an aaa-policy, using the configured authentication-template and authorize-policy configured under aaa-rule. This policy is then added to the explicit proxy.

```

ACOS(config)# aam aaa-policy NEGO_KRB_authen_LDAP_authz
ACOS(config-aaa policy:NEGO_KRB_authen_)# aaa-rule 10

```

```

ACOS(config-aaa policy:NEGO_KRB_authen_-a...)# authentication-template
NEGO_KRB_I
ACOS(config-aaa policy:NEGO_KRB_authen_-a...)# authorize-policy GROUP_
INET
ACOS(config-aaa policy:NEGO_KRB_authen_-a...)# exit
ACOS(config-aaa policy:NEGO_KRB_authen_)# aaa-rule 256
ACOS(config-aaa policy:NEGO_KRB_authen_-a...)# action deny
ACOS(config-aaa policy:NEGO_KRB_authen_-a...)# exit
ACOS(config-aaa policy:NEGO_KRB_authen_)# exit
ACOS(config)# slb virtual-server EP 192.168.91.110
ACOS(config-slb vserver)# port 8084 http
ACOS(config-slb vserver-vport)# service-group EP_DUMMY
ACOS(config-slb vserver-vport)# template policy EP_FORWARD_1
ACOS(config-slb vserver-vport)# aaa-policy NEGO_KRB_authen_LDAP_authz
ACOS(config-slb vserver-vport)# exit
ACOS(config-slb vserver)# exit

```

13. The aaa policies are added to the explicit proxy.

```

ACOS(config)# slb virtual-server EP 192.168.91.110
ACOS(config-slb vserver)# port 8085 http
ACOS(config-slb vserver-vport)# service-group EP_DUMMY
ACOS(config-slb vserver-vport)# template policy EP_FORWARD_1
ACOS(config-slb vserver-vport)# exit
ACOS(config-slb vserver)# port 8086 http
ACOS(config-slb vserver-vport)# service-group EP_DUMMY
ACOS(config-slb vserver-vport)# template policy EP_FORWARD_1
ACOS(config-slb vserver-vport)# exit
ACOS(config-slb vserver)# exit

```

14. Optionally, in the aam authorization policy, the command **forward-policy-authorize-only** may be included as part of the configuration so that the authorization policy provides only server information for configured forward policies, as shown in step 10 of the example.

Explicit Proxy with Virtual Wire Interface

When ACOS uses a virtual wire interface with an explicit proxy deployment, ACOS uses the management interface to resolve the domain name, and uses the virtual wire endpoint at the server side to complete the validation. In this case, the

command **support-cert-fetch** option under **forward-to-proxy** in the SLB template policy is used for certificate fetch traffic to go to the explicit proxy.

The following example shows the **support-cert-fetch** option used to forward certificate fetch traffic to the explicit proxy in a virtual wire, IP-Less configuration.

```
ACOS(config)# slb template policy Explicit_Proxy
ACOS(config-policy)# forward-policy
ACOS(config-policy-forward-policy)# action act_8080
ACOS(config-policy-forward-policy-action)# forward-to-proxy ep8080
support-cert-fetch
ACOS(config-policy-forward-policy-action)# log
ACOS(config-policy-forward-policy-action)# exit
ACOS(config-policy-forward-policy)# exit
ACOS(config-policy)# exit
```

For more information on SSLi deployments and configuration, refer to the SSLi Configuration Guide.

NOTE:

- In a virtual wire IP-less SSLi deployment, it is required to configure DNS and the default gateway for the management interface to resolve the ca-cert domain name.
 - In any typical SSLi deployment, the support-cert-fetch option is supported only when the virtual server is wildcard VIP 0.0.0.0 (transparent proxy).
-

Explicit Proxy with Dual-Stack Forwarding Support

ACOS is enhanced to support IPv4 and IPv6 dual stack forwarding in explicit proxy mode, starting with the DNS resolution. This solution allows the explicit proxy VIP to serve as a gateway between IPv4 (or IPv6) clients and dual-stacked servers, giving preference to IPv6.

Prior to this feature, although the dual-stacked operation was already supported in ACOS through DNS dynamic service templates as well as separate SLB policy templates, it lacked dynamic address translation or interconnectivity, which could lead to service outages or delays in some cases. For instance, you could configure

IPv4 and IPv6 DNS servers on the same dynamic service template, but an IPv4 explicit proxy VIP serving an IPv4 client could only reach out to remote servers over IPv4. Similarly, an IPv6 explicit proxy VIP serving an IPv6 client could only reach out to remote servers over IPv6.

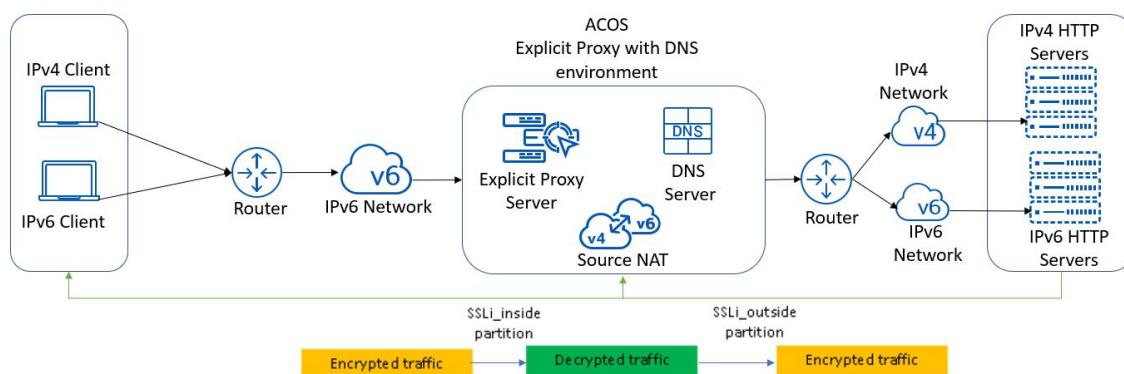
With this feature, once you configure IPv4 and IPv6 DNS servers under the dynamic service template, it behaves as follows:

- An IPv6 DNS server is given preference over an IPv4 server to send DNS requests.
- AAAA (IPv6) and A (IPv4) record requests are sent to the DNS server as two separate requests unless a cached record exists.
- AAAA record replies are preferred over A records.
- Based on the DNS response, the appropriate source NAT pool i.e., either IPv6 NAT-pool or IPv4 NAT-pool is used.

This enables an explicit proxy virtual port to forward HTTP or HTTPS requests via IPv4 or IPv6 networks. Additionally, when this feature is configured, the SSLi certificate fetching module also allows connections with IPv4 addresses on the client side and IPv6 addresses on the server side and vice versa.

NOTE: This feature is not applicable to Transparent Proxy and Proxy chaining.

Deployment



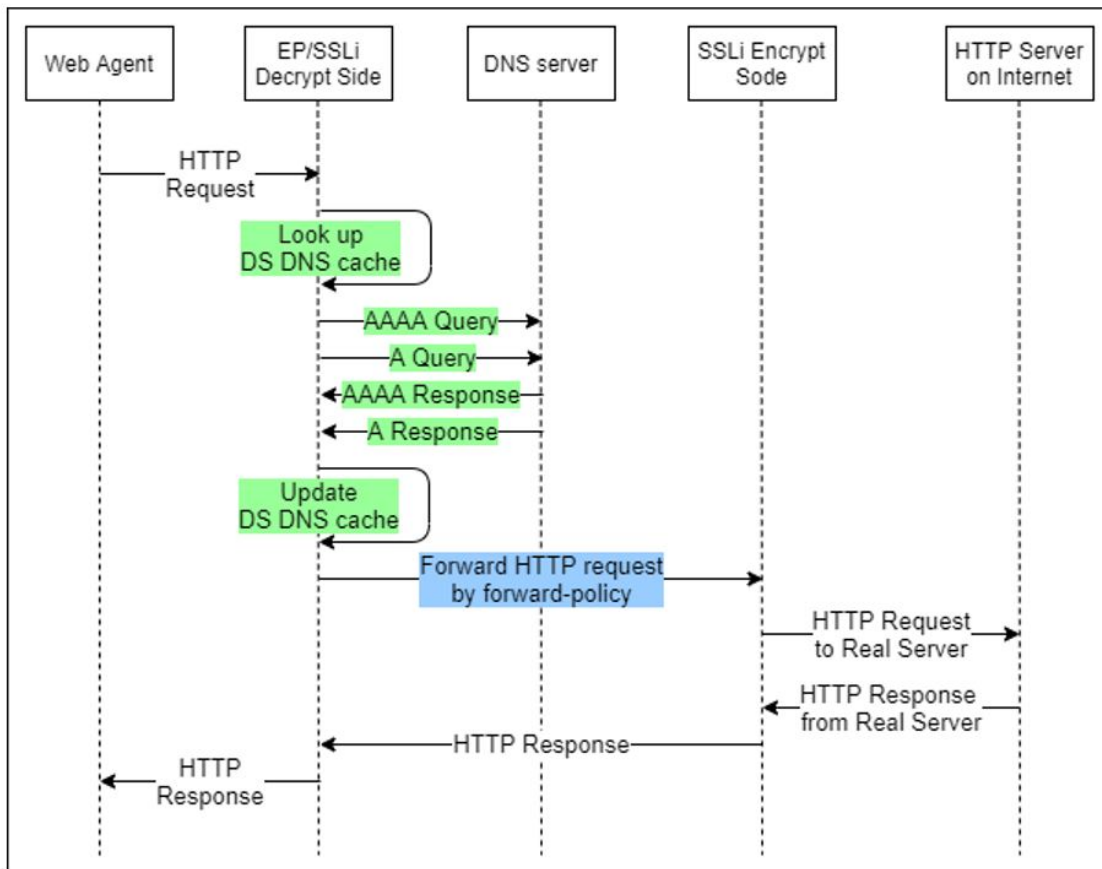
For dual-stack support, the following configuration updates are needed in the existing explicit proxy environment:

1. Create or modify the SLB template dynamic service to configure IPv4 and IPv6 DNS servers on the same template. You can define a maximum of two default DNS servers and 64 class-lists with each class-list containing two DNS server IPs within each dynamic-service template.
2. Create or modify the SLB policy template to enable dual-stack forward policy action and bind the action to any destination rule of the forward policy.
3. Configure IPv4 and IPv6 connectivity using appropriate VIPs, NAT pools, and service groups.
4. Create or modify the SSLi certificate fetching module to support connections with IPv4/IPv6 at the client as well as server side.

For more information on SSLi deployments and configuration, see *SSLi Configuration Guide*.

Traffic Workflow

The following figure illustrates the traffic workflow for this configuration.



1. After the Explicit Proxy VIP (IPv4 or IPv6) receives the client's HTTP request (over IPv4 or IPv6), a DNS lookup is first performed in the DNS cache.
2. If no records are found, the dynamic service module sends two separate DNS requests, one for AAAA and one for A record to the DNS server.
3. The DNS server responds with an AAAA type and A type record (in that order).
4. The DNS response is updated in the DNS cache.
5. Based on the DNS response, the explicit proxy uses the appropriate source NAT pool i.e., either an IPv4 NAT pool or an IPv6 NAT pool to forward the requests. However, preference is given to AAAA records over A records.

CLI Configuration

In the topology shown in the above figure, the client device is connected to the Explicit Proxy SSLi solution, which is then connected to the external gateway. The

ACOS device is connected to the client with a partition called **ACOS_decrypt**. The ACOS device is also connected to the external gateway with a partition called **ACOS_encrypt**.

The following are the required configurations in ACOS_decrypt partition:

1. Create an SLB dynamic service template and specify default IPv4 and IPv6 Internet DNS resolvers.

```
ACOS(config)# slb template dynamic-service DNS
ACOS(config-dynamic-service)# dns server 10.10.10.27
ACOS(config-dynamic-service)# dns server 10.1.1.127
```

Alternatively, you can also specify a class-list with specific DNS server IPs for directing traffic to particular destination domains.

```
ACOS(config)# slb template dynamic-service DNS
ACOS(config-dynamic-service)# class-list test priority 30
ACOS(config-dynamic-service-class-list)# dns server 10.2.2.140
ACOS(config-dynamic-service-class-list)# dns server 10.10.20.440
```

2. Configure source NAT pools for the IPv4 service group, IPv6 service group, and IPv6 fallback service group.\

Additionally, enable NAT Application-Layer Gateway (ALG) support for the Point-to-Point Tunneling Protocol (PPTP).

```
ACOS(config)# ip nat pool EP_POOL_v4 5.5.5.100 5.5.5.100 netmask /32
gateway 5.5.5.177
ACOS(config)# ipv6 nat pool EP_POOL_v6 5001::1:50 5001::1:50 netmask
128 gateway 5001::1:171
ACOS(config)# ip nat alg pptp enable
ACOS(config)# ipv6 nat pool fall_back_v6 5001::1:175 5001::1:175
netmask 128 gateway 5001::1:11
```

3. Create real servers and service groups for IPv4 and IPv6 networks.

```
ACOS(config)# slb server Internet_v4_Server 5.5.5.5
ACOS(config-real server)# health-check-disable
ACOS(config-real server)# port 80 tcp
ACOS(config-real server-node port)# health-check-disable
ACOS(config-real server-node port)# exit
ACOS(config-real server)# port 443 tcp
```

```

ACOS(config-real server-node port)# health-check-disable
ACOS(config-real server-node port)# exit

ACOS(config)# slb server Internet_v6_server 7001::1:10
ACOS(config-real server)# health-check-disable
ACOS(config-real server)# port 80 tcp
ACOS(config-real server-node port)# health-check-disable
ACOS(config-real server-node port)# exit
ACOS(config-real server)# port 443 tcp
ACOS(config-real server-node port)# health-check-disable
ACOS(config-real server-node port)# exit

ACOS(config)# slb service-group To_Internet_v4 tcp
ACOS(config-slb svc group)# health-check-disable
ACOS(config-slb svc group)# member Internet_v4_Server 80
ACOS(config-slb svc group-member:80)# exit
ACOS(config-slb svc group)# exit
ACOS(config)# slb service-group To_internet_v6 tcp
ACOS(config-slb svc group)# health-check-disable
ACOS(config-slb svc group)# member Internet_v6_server 80
ACOS(config-slb svc group-member:80)# exit
ACOS(config-slb svc group)# exit

```

4. Configure the client-SSL template to enable SSLi (forward-proxy).

```

ACOS(config)# slb template client-ssl SSLInsight_ClientSide
ACOS(config-client ssl)# forward-proxy-ca-certificate Cert123.pem key
key123456
ACOS(config-client ssl)# forward-proxy-enable

```

5. Configure the forward policy with dual-stack action.

Specify the service groups and source NAT pools for IPv4 and IPv6 networks. Also, specify the fallback service group and corresponding source NAT pool to handle requests when a domain name is not resolved. The unresolved requests are forwarded to the fallback service group.

Bind the dual-stack action to any destination rule of the forward policy template.

```

ACOS(config)# slb template policy DUAL_Stack_6n4
ACOS(config-policy)# forward-policy
ACOS(config-policy-forward-policy)# dual-stack-action FWD_6n4

```



```

ACOS(config-policy-forward-policy-dual-st...)# ipv4 To_Internet_v4 snat
EP_POOL_v4
ACOS(config-policy-forward-policy-dual-st...)# ipv6 To_Internet_v6 snat
EP_POOL_v6
ACOS(config-policy-forward-policy-dual-st...)# fallback To_Internet_v6
snat fall_back_v6
ACOS(config-policy-forward-policy-dual-st...)# log
ACOS(config-policy-forward-policy-dual-st...)# exit
ACOS(config-policy-forward-policy)# source ANY
ACOS(config-policy-forward-policy-source)# match-any
ACOS(config-policy-forward-policy-source)# destination any dual-stack-
action FWD_6n4

```

6. Create an explicit proxy virtual server and bind the dual-stack template policy (configured in step 5) to it.

In this configuration, *EP_VIP* is an IPv4 virtual server that functions as a dual-stacked HTTP/HTTPS explicit proxy for IPv4 clients.

```

ACOS(config)# slb virtual-server EP_VIP 10.10.10.150
ACOS(config-slb vserver)# port 3128 http
ACOS(config-slb vserver-vport)# support-http2
ACOS(config-slb vserver-vport)# template policy DUAL_Stack_6n4
ACOS(config-slb vserver-vport)# source-nat auto
ACOS(config-slb vserver-vport)# service-group To_Internet_v4
ACOS(config-slb vserver-vport)# template dynamic-service DNS
ACOS(config-slb vserver-vport)# template client-ssl SSLInsight_
ClientSide

```

In this configuration, *EP_VIP_v6* is an IPv6 virtual server that functions as a dual-stacked HTTP/HTTPS explicit proxy for IPv6 clients.

```

ACOS(config)# slb virtual-server EP_VIP_v6 1001::1:150
ACOS(config-slb vserver)# port 3128 http
ACOS(config-slb vserver-vport)# support-http2
ACOS(config-slb vserver-vport)# template policy DUAL_Stack_6n4
ACOS(config-slb vserver-vport)# source-nat auto
ACOS(config-slb vserver-vport)# service-group To_internet_v6
ACOS(config-slb vserver-vport)# template dynamic-service DNS
ACOS(config-slb vserver-vport)# template client-ssl SSLInsight_
ClientSide

```

Explicit Proxy Without SSLi

In Explicit Proxy (EP) deployments, ACOS forwards client traffic to an upstream proxy based on the configured forward-policy. By default, the destination port of the forwarded traffic is derived from the virtual port configured on the ACOS device.

In EP deployments where SSLi is not configured (no `client-SSL template`) and HTTP/2 is used, the upstream proxy destination port may not be automatically translated if it differs from the virtual port. This can lead to forwarding traffic with an incorrect destination port.

To explicitly set the destination port for traffic forwarded to the upstream proxy, you can use the `force-dst-port` option under the `forward-policy action` to enforce the correct destination port for all upstream proxy connections.

This feature is useful in the following scenarios:

- Explicit proxy chaining without SSLi
- Deployments using HTTP/2
- Environments where the virtual port and upstream proxy port are different
- Service chaining architectures requiring precise port control.

CLI Configuration

The following example shows how to configure `force-dst-port` so that traffic is forwarded to the correct upstream proxy port:

1. Configure a source NAT pool for connections from ACOS to the upstream proxy.

```
ip nat pool UPSTREAM_PROXY_SNAT 172.92.222.245 172.92.222.245 netmask
/32
!
```

2. Configure the upstream proxy server and its listening port.

```
slb server UPSTREAM_PROXY 172.92.86.86
port 3128 tcp
!
```

3. Configure the service group for the upstream proxy.

```
slb service-group SG_UPSTREAM_PROXY tcp
  member UPSTREAM_PROXY 3128
!
```

4. Create a policy template and configure a forward policy.

```
slb template policy EPC
  forward-policy
    enable-adv-match
    source EPC_ANY_0
    match-any
    destination any action TO_UPSTREAM_PROXY
      forward-to-proxy SG_UPSTREAM_PROXY snat UPSTREAM_PROXY_SNAT
      force-dst-port 3128
    log
!
```

5. Configure the EP virtual server and apply the policy template.

```
slb virtual-server vs_epc 172.91.222.145
  port 3129 http
  template policy EPC
  source-nat auto
  service-group SG_UPSTREAM_PROXY
!
```

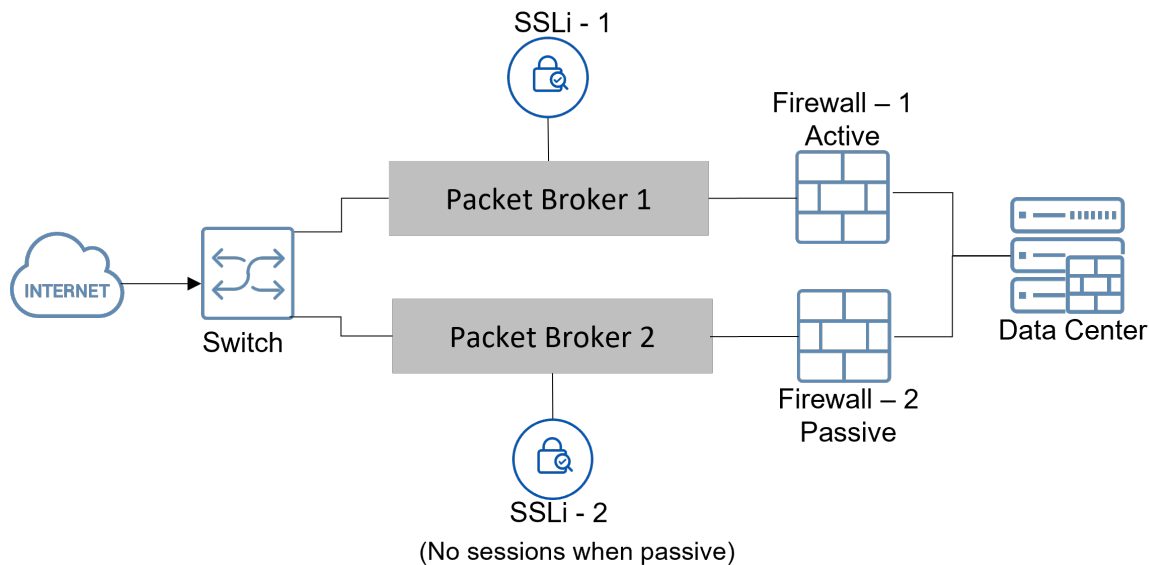
The **force-dst-port** option forces ACOS to use the configured destination port when forwarding traffic to the upstream proxy. This is useful when the Explicit Proxy virtual port and the upstream proxy listening port are different, for example, EP virtual port 3129 forwarding to upstream proxy port 3128.

Terminating SSLi Session Silently

When SSLi terminates idle sessions, it sends out RST and FIN packets in both directions, i.e., to the client and the server. This ensures that all sessions between clients, servers, and ACOS are always gracefully terminated. However, this behavior can be problematic in pure Layer-2 SSLi deployments with redundant paths, when the upstream and downstream networking devices, such as firewalls, govern the active-standby paths. See [Figure 85](#).

In such deployments, the upstream firewall selects an active path, and sessions are established on the active SSLi appliance. When a failover occurs to the standby path, gratuitous ARPs are sent out, and sessions start re-establishing on the standby SSLi appliance. However, in due time, the previously active SSLi appliance sends RSTs and FINs to terminate stale sessions, but this disrupts the firewall's MAC address tables, resulting in network loops.

Figure 85 : Layer 2 SSLi Deployment



To address this behavior, ACOS now supports silently terminating stale SSLi sessions so that the passive SSLi appliance does not send out any RSTs or FINs.

This feature supports both HTTP/1.1 and HTTP/2 connections.

NOTE: Enabling this feature may cause sessions to stall on either the client or the server side.

CLI Configuration

To enable the feature, use the `ssli-silent-termination-enable` command (per partition).

```

ACOS(config)# slb common
ACOS(config-common)# ssli-silent-termination-enable
  
```

When this command is enabled and a client-SSL or server-SSL template is bound to the virtual port, the RST or FIN packets are not sent in the following scenarios:

- Session timeout
- Session timeout with close notify enabled
- `clear session` command is triggered.

GUI Configuration

1. Navigate to **ADC >> SLB >> Global** tab.
2. Enable the 'Enable SSLi Silent Termination' option.

Transparent Proxy with Normal VIPs Associated to ACL

This section describes the transparent proxy with normal VIP associated to ACL, its features, configuration, and limitations.

The following topics are covered:

Overview

ACOS can be configured to associate normal VIPs with ACLs. The destination IP in ACL can be transformed to the VIP, and the VIP bound to an ACL. The source IP defined in the associated ACL will continue to determine if the traffic matches the specified virtual port and accordingly permit or deny the traffic. You can continue to configure wildcard VIPs associated with ACLs along with the normal VIP and ACL configuration.

Key Considerations

- The ACL associated to the VIP should not contain any destination IP.
- Both IPv4 and IPv6 addresses can be configured.
- If ACOS is unable to match the traffic with the virtual port in one VIP with ACL, it tries to match the virtual port in another VIP with ACL.

- If ACOS is unable to match the traffic with any virtual port in the VIPs, it tries to match the virtual port in the wildcard VIP.
- After creating a normal VIP with ACL, you can modify the IP address and ACL. However, you cannot remove the associated ACL after it is created.
- Multiple VIPs can use the same IP address if they are associated with ACLs. If there are multiple VIPs with ACL configured with the same IP in the system, incoming traffic will match only one of the VIP and will be rejected by other VIPs.
- The VIP associated to ACL can have the same IP address as the IP of the SLB server.
- The VIP associated to ACL cannot use the same IP as a normal VIP and subnet VIP.
- An ACL can be associated with two virtual servers, only if the two virtual servers have different VIPs.
- If a virtual port is matched by IP and L4 port but is rejected by the ACL bound in a virtual server, ACOS skips the virtual port and tries to match the next virtual port.
- The VIP associated to ACL uses the same resource pool as a normal virtual server. So, like virtual servers, the maximum number of configurable VIPs depends on the platform and system memory size and is greater than the maximum configurable number of wildcard VIPs.

Limitations

This feature has the following limitations:

- It is only supported for slow paths.
- Subnet VIPs cannot be associated with an ACL.
- The matching order for VIPs with ACL having the same IP address is not guaranteed. The matching does not follow the alphabetic order of ACL, the way it is done in wildcard VIPs.

Basic Configuration

This section describes the guidelines for configuring a normal VIP associated to an ACL in transparent proxy deployment. The VIP acts as a non-wildcard VIP bound to an ACL and determines the traffic destination.

1. Enable ethernet interfaces.

```
ACOS(config)# interface ethernet 1
ACOS(config-if:ethernet:1)# enable
ACOS(config-if:ethernet:1)# ip address 192.105.60.10 255.255.255.0
ACOS(config-if:ethernet:1)# exit
ACOS(config)# interface ethernet 2
ACOS(config-if:ethernet:2)# enable
ACOS(config-if:ethernet:2)# ip address 152.145.30.11 255.255.255.0
```

2. Configure ACLs for permitting or denying traffic from specific source IP addresses. For example:

```
ACOS(config)#ip access-list myacl1
ACOS(config-ext-access-list:myacl1)# permit ip 152.145.30.5 0.0.0.0 any
ACOS(config-ext-access-list:myacl1)# deny ip 152.145.30.6 0.0.0.0 any
ACOS(config-ext-access-list:myacl1)# exit

ACOS(config)#ip access-list myacl2
ACOS(config-ext-access-list:myacl2)# permit ip 152.145.30.6 0.0.0.0 any
ACOS(config-ext-access-list:myacl2)# deny ip 152.145.30.5 0.0.0.0 any
ACOS(config-ext-access-list:myacl2)# exit

ACOS(config)#ip access-list myacl3
ACOS(config-ext-access-list:myacl3)# permit ip 152.145.30.7 0.0.0.0 any
ACOS(config-ext-access-list:myacl3)# deny ip 152.145.30.6 0.0.0.0 any
ACOS(config-ext-access-list:myacl3)# exit
```

3. Configure a real server.

```
ACOS(config)# slb server sw 152.145.30.1
ACOS(config-real server)# port 80 tcp
ACOS(config-real server-node port)# exit
```

4. Configure a service-group.

```
ACOS(config)# slb service-group sw-grp tcp
ACOS(config-slb svc group)# member sw 80
ACOS(config-slb svc group-member:80)# exit
```

5. Configure virtual servers and virtual ports. Associate the virtual servers with a VIP associated to an ACL.

```
ACOS(config)# slb virtual-server vs1 0.0.0.0 acl name myacl3
ACOS(config-slb vserver)# port 80 tcp
ACOS(config-slb vserver-vport)# service-group sw-grp
ACOS(config-slb vserver-vport)# no-dest-nat
ACOS(config-slb vserver)# exit

ACOS(config)# slb virtual-server vs2 152.145.30.1 acl name myacl1
ACOS(config-slb vserver)# port 80 tcp
ACOS(config-slb vserver-vport)# service-group sw-grp
ACOS(config-slb vserver-vport)# no-dest-nat
ACOS(config-slb vserver)# exit

ACOS(config)# slb virtual-server vs3 152.145.30.1 acl name myacl2
ACOS(config-slb vserver)# port 80 tcp
ACOS(config-slb vserver-vport)# service-group sw-grp
ACOS(config-slb vserver-vport)# no-dest-nat
ACOS(config-slb vserver)# exit
```

Consolidated Configuration Example

This section describes the consolidated configuration of IP-based VIPs associated to ACLs.

```
ACOS(config)#show running-config
!Current configuration: 2015 bytes
!Configuration last updated at 10:09:08 GMT Wed Dec 6 2023
!Configuration last saved at 10:09:12 GMT Wed Dec 6 2023
!64-bit Advanced Core OS (ACOS) version 6.0.3
!
ip access-list myacl1
 4 permit ip 152.145.30.5 0.0.0.0 any
 8 deny ip 152.145.30.6 0.0.0.0 any
!
ip access-list myacl2
 4 permit ip 152.145.30.6 0.0.0.0 any
 8 deny ip 152.145.30.5 0.0.0.0 any
!
ip access-list myacl3
 4 permit ip 152.145.30.7 0.0.0.0 any
 8 deny ip 152.145.30.6 0.0.0.0 any
!
```



```
interface ethernet 1
  enable
  ip address 192.105.60.10 255.255.255.0
!
interface ethernet 2
  enable
  ip address 152.145.30.11 255.255.255.0
!
slb server sw 152.145.30.1
  port 80 tcp
!
slb service-group sw-grp tcp
  member sw 80
!
slb virtual-server vs1 0.0.0.0 acl name myacl3
  port 80 tcp
  service-group sw-grp
  no-dest-nat
!
slb virtual-server vs2 152.145.30.1 acl name myacl1
  port 80 tcp
  service-group sw-grp
  no-dest-nat
!
slb virtual-server vs2 152.145.30.1 acl name myacl2
  port 80 tcp
  service-group sw-grp
  no-dest-nat
!
```

Explicit FTP Proxy

This section provides an overview of explicit FTP proxy, configuration resources that are available, and how to configure it.

The following topics are covered:

Overview	879
Configuring Explicit FTP Proxy	880

Overview

An ACOS device can be configured to provide explicit FTP proxy services. It operates similar to Explicit HTTP Proxy ([HTTP Proxy](#)). Through a policy template, it manages client access to hosts based on lists of allowed traffic sources (clients) and destinations (FTP servers). It also provides these services in an SSLi configuration (See *SSL Configuration Guide* for SSL and SSLi configuration). An aaa-policy can be configured for authentication, but is not required.

When this feature is enabled, an FTP virtual port on the ACOS device intercepts FTP requests from clients, validates both the sources and the destinations, and forwards only those requests that come from valid sources and that are sent to approved destinations. Destinations are validated based on layer 3 IP address, and hostname strings. For approved destinations, the ACOS device performs DNS lookups to obtain the IP addresses.

The destinations requested by clients can be filtered based on the hostname in the request's Host header.

If both the source and destination are allowed, ACOS translates the client address into a NAT address, if applicable, and forward the request to the destination.

If either the source or the destination is not explicitly allowed by the applicable source or destination list, the request is dropped.

There are three mechanisms by which explicit FTP proxy can forward traffic: traffic may be forwarded to the Internet, to a specified service group, or to another explicit proxy server.

Features

Explicit FTP Proxy differs from Secure FTP proxy (see [Secure FTP Proxy](#)) in the following ways:

Secure FTP proxy

- FTP client provides only one pair of username/password for the FTP server.
- FTP client does not provide real FTP server address through FTP command

Explicit FTP Proxy

- FTP client can provide two username/password pairs, one for the FTP proxy and one for the FTP server.
- FTP client provides the real FTP server address with FTP commands (SITE/OPEN/USER...)
- Works with web-category-list
- Active FTP support

Known Limitations

ACOS can be configured as an FTP load balancer (see [FTP Load Balancing](#)) or as an explicit FTP proxy. It cannot function as both. Explicit SSL/TLS FTP only works with passive mode.

Configuring Explicit FTP Proxy

Configuration Resources for Explicit FTP Proxy

To provide precise control, a forward policy defined in the SLB policy template is used to identify the sources, destinations, and actions for matching traffic. [Table 21](#) describes these resources and how they are related.

Table 21 : Configuration Resources for Explicit FTP Proxy

Resource	Purpose	Description
Policy template	Applies actions to client-to-server traffic based on source and destination	Set of rules that define permitted traffic sources and destinations, and actions to apply to permitted traffic. This is defined within the SLB command. The explicit ftp proxy relies only on the forward policy to define the source and destination match criteria, and corresponding actions to apply. For more information, see the forward policy description.

Table 21 : Configuration Resources for Explicit FTP Proxy

Resource	Purpose	Description
Forward Policy	Specifies permitted traffic destinations and sources	Collection of source and destination match rules based on source IP addresses, destination hosts of client requests, and actions to apply. Forward policy is a required component when configuring explicit FTP proxy. Through this module, you can define the type and scope of the action desired for outbound traffic. The <code>no-client-reuse</code> command is ignored by explicit FTP proxy.
Source Match Rule	Specify sources (or clients) IP addresses used to match a forward policy rule	Source list specifies an IPv4 and/or IPv6 class list used to match a set of clients. Each entry in a source class list specifies an IPv4 or IPv6 address. Use the <code>match-class-list</code> command to match the source IP. The <code>match-any</code> command is ignored if configured under <code>ftp-proxy</code> .
Destination Match Rule	Specifies a destination match rule, corresponding action to apply when there is a match, and the priority of the rule	When specifying a class-list or web-category-list for destination matching, the following parameters are available: host - ftp proxy will check the real FTP server's hostname URL - ftp proxy ignores this parameter as URL is not applicable. IP - If a class-list is specified, the IP parameter option is available. If the destination rule matches, the action specified in the rule is applied. The priority of the rule is used to break ties when multiple destination rules are matched. The higher priority value takes precedence.
Action	Defines the	Actions define what to do with the request and

Table 21 : Configuration Resources for Explicit FTP Proxy

Resource	Purpose	Description
	action to take corresponding to the matching rule	whether to log the event. The scope of the action object is within the forward policy. The action can be one of the following: <code>forward-to-internet</code>, <code>forward-to-service-group</code>, <code>forward-to-proxy</code>, <code>log</code>, or <code>drop</code>. Source NAT is an optional action that may be applied. In the case of an upstream proxy server and an SSLi configuration, <code>forward-to-proxy</code> is used instead of <code>forward-to-internet</code> or <code>forward-to-service-group</code>.
Explicit FTP-proxy port	Intercepts FTP requests from clients	Virtual port that receives FTP requests from traffic sources. This configuration resource consists of a real server configuration, a service group, a virtual server, and an FTP-proxy virtual port. If using <code>forward-to-internet</code> as a forwarding mechanism, configure an “Internet” real server and service-group. This can act as a placeholder configuration which represents the “Internet” rather than an actual real device. This is an essential component to ensure feature functionality. Every allowed server port to the Internet must be configured in the real server and service group. For example, typically port 21 is used for FTP traffic. Ports must be configured in the service group along with a real server representing the Internet.
Dynamic Service Template	Provides DNS resolution when forwarding to Internet	Forward to Internet traffic requires DNS resolution. When the <code>forward-to-internet</code> command is used, you must define one or more default DNS servers and /or class-lists with specific DNS domain names and DNS servers in a dynamic service template to resolve host names

Table 21 : Configuration Resources for Explicit FTP Proxy

Resource	Purpose	Description
		to IP addresses.
Real Server (s) and service group	Locally serves content if destination cannot be reached	Standard SLB configuration with real servers and service group, to be called in Forward Policy
VIP with FTP vPort	Receives requests to be served locally	
NAT Resources		
The following resources are required only if sources (clients) require source NAT.		
NAT pool	Assigns outside addresses to inside clients	Range of public (externally routable) IP addresses to assign to clients before forwarding the client traffic to the Internet.
AAM Resources		
The following resources are required only for authentication through aaa-policy.		
AAA-policy	Authenticates the user through use of aam authentication server bound to authentication-template.	Supports LDAP, RADIUS, and WINDOWS (KERBEROS). Bind the aaa-policy at the virtual port configuration level. Explicit FTP proxy only checks the aaa-policy for authentication server configuration information. If an aaa-policy is assigned in the virtual port, the explicit FTP proxy expects the client to provide the username/password for the proxy. If aaa-policy is not configured, the explicit FTP proxy waits for the SITE/OPEN/USER command, then connects to the FTP server.

Basic network resources, including network interface connections to the sources and destinations and DNS servers, are also required.

FTP Proxy Configuration Procedures

Basic FTP Proxy Configuration Procedure

1. Configure a real FTP server
2. Configure port with port number and service type as TCP
3. Configure service-group for real FTP server(s) and add to the group as members
4. Configure virtual server
5. Configure virtual port with port number and service type as ftp-proxy

For SSL Offload, bind the client-ssl template at the virtual port configuration level.

Explicit FTP Proxy Procedure

1. Configure policy template
2. Configure AAA-policy
3. Configure dynamic service template
4. Bind the policy, dynamic-service templates and aaa-policy at the virtual port sub-configuration level.

FTP Proxy Configuration Examples

Basic FTP Proxy Configuration (CLI Example)

1. Create an access list to use as part of an aaa-rule in aam aaa-policy my-aaa-policy.

```
ACOS(config)# access-list 2 permit 192.0.2.0 0.0.0.0
```

2. Create a source class list of IPv4 type. Source class lists may be of IPv4 or IPv6 types and are defined as a list of IP subnets and network masks.

```
ACOS(config)# class-list cl-internet ipv4
```

```
ACOS(config-class list)# 192.0.2.0/32
```

```
ACOS(config-class list)# exit
```

3. Create a web-category-list to apply as part of source-destination configuration for policy.


```
ACOS(config)# web-category
ACOS(config-web-category)# category-list Wrong
ACOS(config-web-category-category-list)# hacking
ACOS(config-web-category-category-list)# illegal
ACOS(config-web-category-category-list)# exit
ACOS(config-web-category)# exit
```

4. Configure the FTP server and service group to handle requests.

```
ACOS(config)# slb server ftp-server 198.51.100.0
ACOS(config-real server)# port 21 tcp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
ACOS(config)# slb service-group sg tcp
ACOS(config-slb svc group)# member ftp-server 21
ACOS(config-slb svc group-member:21)# exit
ACOS(config-slb svc group)# exit
```

5. Configure actions for source matching.

```
ACOS(config)# slb template policy policy-template
ACOS(config-policy)# forward-policy
ACOS(config-policy-forward-policy)# action drop
ACOS(config-policy-forward-policy-action)# drop
ACOS(config-policy-forward-policy-action)# exit
ACOS(config-policy-forward-policy)# action act-internet
ACOS(config-policy-forward-policy-action)# forward-to-internet sg snat
NAT
ACOS(config-policy-forward-policy-action)# exit
ACOS(config-policy-forward-policy)# action act-sg
ACOS(config-policy-forward-policy-action)# log
ACOS(config-policy-forward-policy-action)# forward-to-service-group sg
snat NAT
ACOS(config-policy-forward-policy-action)# exit
```

6. Configure the source match rules and actions to apply based on the match and priority.

```
ACOS(config-policy-forward-policy)# source src-internet
ACOS(config-policy-forward-policy-source)# match-class-list cl-internet
ACOS(config-policy-forward-policy-source)# destination any action act-
internet
```

```

ACOS(config-policy-forward-policy-source)# exit
ACOS(config-policy-forward-policy)# source bad-dest
ACOS(config-policy-forward-policy-source)# destination web-category-
list Wrong action drop host priority 100
ACOS(config-policy-forward-policy-source)# exit
ACOS(config-policy-forward-policy)# source src-sg
ACOS(config-policy-forward-policy-source)# match-any
ACOS(config-policy-forward-policy-source)# destination any action act-
sg
ACOS(config-policy-forward-policy-source)# exit
ACOS(config-policy-forward-policy)# exit
ACOS(config-policy)# exit

```

7. Configure the aam authentication servers. An LDAP and RADIUS server are configured in this example.

```

ACOS(config)# aam authentication server ldap ldap
ACOS(config-ldap auth server:ldap)# host 203.0.113.72
ACOS(config-ldap auth server:ldap)# base ou=People,dc=mdept,dc=com
ACOS(config-ldap auth server:ldap)# dn-attribute uid
ACOS(config-ldap auth server:ldap)# exit
ACOS(config)# aam authentication template ldap-ex
ACOS(config-auth template:ldap-ex)# server ldap
ACOS(config-auth template:ldap-ex)# exit
ACOS(config)# aam authentication server radius rad-svr2
ACOS(config-radius auth server:rad-svr2)# host 203.0.113.71
ACOS(config-radius auth server:rad-svr2)# secret example
ACOS(config-radius auth server:rad-svr2)# exit
ACOS(config)# aam authentication template rad-ex
ACOS(config-auth template:rad-ex)# server rad-svr2
ACOS(config-auth template:rad-ex)# exit

```

8. Create a dynamic service template and specify Internet DNS resolver.

```

ACOS(config)# slb template dynamic-service ds
ACOS(config-dynamic-service)# dns server 203.0.113.50
ACOS(config-dynamic-service)# exit

```

9. Create the aam aaa-policy which contains the authentication servers that were created.

```

ACOS(config)# aam aaa-policy my-aaa-policy

```

```

ACOS(config-aaa policy:my-aaa-policy)# aaa-rule 1
ACOS(config-aaa policy:my-aaa-policy-aaa ...)# access-list 2
ACOS(config-aaa policy:my-aaa-policy-aaa ...)# authentication-template
ldap-ex
ACOS(config-aaa policy:my-aaa-policy-aaa ...)# exit
ACOS(config-aaa policy:my-aaa-policy)# aaa-rule 2
ACOS(config-aaa policy:my-aaa-policy-aaa ...)# authentication-template
rad-ex
ACOS(config-aaa policy:my-aaa-policy-aaa ...)# exit
ACOS(config-aaa policy:my-aaa-policy)# exit

```

10. Bind the aaa-policy, template policy, and dynamic service templates to the ftp-proxy virtual port.

```

ACOS(config)# slb virtual-server ftp-vip 192.168.91.105
ACOS(config-slb vserver)# port 21 ftp-proxy
ACOS(config-slb vserver-vport)# source-nat auto
ACOS(config-slb vserver-vport)# service-group sg
ACOS(config-slb vserver-vport)# aaa-policy my-aaa-policy
ACOS(config-slb vserver-vport)# template policy policy-template
ACOS(config-slb vserver-vport)# template dynamic-service ds
ACOS(config-slb vserver-vport)# exit
ACOS(config-slb vserver)# exit

```

Basic FTP Proxy Configuration (GUI Example)

Configure a real FTP server

1. Navigate to **ADC>>SLB**.
2. Click the **Servers** tab.
3. Click **Create**.
4. Enter the name “ftp-server” in the Name field.
5. Enter host “198.51.100.0” in the Host field.

Configure port with port number and service type as ftp

1. In the Port section, click **Create**.
2. In the Port Number field, enter the port number “21”.
3. In the Protocol drop-down list, select TCP.

4. Click **Create**.
5. Click Update (to complete server creation process).

Configure service-group for real FTP server(s) and add to the group as members

1. From **ADC>>SLB**, click the **Service Groups** tab.
2. Click **Create**.
3. Enter the name “sg” in the Name field.
4. In the Member section, click **Create**.
5. In the Server drop-down list, select the ftp-server “ftp-server”.
6. In the port field, enter the port number “21”.
7. Click **Create**.

Configure virtual server

1. From **ADC>>SLB**, click the **Virtual Servers** tab.
2. Click **Create**.
3. Enter the name “ftp-vip” for the virtual server in the Name field.
4. Enter the IP address “192.168.91.105” for the virtual server in the IP Address field.

Configure virtual port with port number and service type as ftp-proxy.

1. In the Virtual Port section, click **Create**.
2. In the Protocol drop-down list, select FTP Proxy.
3. In the Port field, enter the port “21”.
4. In the Service Group field, select the service group “sg”.
5. Click **Create**.
6. For SSL Offload, bind the client-ssl template at the virtual port configuration level.
7. See the *SSL Insight Configuration Guide* for information about SSL configuration.
8. Click **Edit** for the newly created Virtual Port.
9. Expand the Templates section by clicking on +.

10. In the Template Client SSL drop-down list, select an existing template, or click **+** to create a new one.
11. Click **Update** when configuration is done.
12. Click **Update** to finish configuring the virtual server.

Explicit FTP Proxy Template Configuration and Binding (GUI Example)

Configure policy template

1. Navigate to **Security>>Forward Proxy** and click the **Templates** tab.
2. Click **Create** and select Policy.
3. In the Name field, enter the name “policy-template” for the policy template.
4. With the Action Policies tab selected, click **Add**.
5. In the Name field, enter the name of the Action.
6. In the Action drop-down list, select the Action to take when source policies are applied.
7. When configuration is done, click **OK**.
8. Click the **Source Policies** tab.
9. Click **Add**.
10. In the Name field, enter the name of the Source Policy.
11. Click **Add** to configure Destination Rules.
12. After configuration is complete, click **OK**.

Configure AAA-policy

1. Navigate to **AAM>>Policies>>Templates**.
2. Click the **Authentication Templates** tab.
3. Click **Create**.
4. In the Name field, enter the name “ldap-ex” for the authentication template.
5. Click **New Authentication Server**.
6. Select the Authentication Server type “LDAP” from the Type drop-down list and click **Next**.

7. In the Name field, enter the name “ldap” for the authentication server.
8. Configure any additional settings and click **Create**.
9. Click **Update**.
10. Click the **AAA Policies** tab.
11. Click **Create**.
12. In the Name field, enter the name “my-aaa-policy” for the AAA policy.
13. In the Virtual Port Bindings list, select the virtual server/virtual port, “ftp-vip 21+ftp-proxy”.
14. Click **Create**.
15. In the AAA Rules section, click Create to add AAA rules to the AAA policy.
16. To create access lists for use with AAA Rules, Navigate to **Security>>Access List**.
17. Click **Create** when done with configuring AAA rules.
18. Click **Update**.

Configure dynamic service template

1. Navigate to **Security>>Forward Proxy**, and click the **Templates** tab.
2. Click **Create** and select Dynamic Service.
3. In the Name field, enter the name “dynamic-service” for the dynamic service template.
4. In the IPv4 DNS Server 1 field, enter “203.0.113.50” for the DNS server.
5. Click **OK**.

Bind the policy template and dynamic service template to the virtual port.

1. Navigate to **ADC>>SLB** and click the **Virtual Servers** tab.
2. Click **Edit** for “vip-ftp”, our virtual server.
3. In the Virtual Port section, click **Edit** on the configured virtual port.
4. Expand the Templates section by clicking on +.
5. In the Template Policy list, select the “policy-template” to bind to the virtual port.
6. In the Dynamic Service list, select “dynamic-service” to bind to the virtual port.

7. Click **Update** to complete configuration update to virtual port.
8. Click **Update** to complete configuration update to virtual server.

Displaying and Clearing SLB FTP Proxy Statistics

The following shows steps from the CLI and GUI to show ftp-proxy statistics and clear the counters.

Displaying FTP Proxy Statistics (CLI Example)

The following example shows output from the `show slb ftp-proxy` command.

```
ACOS# show slb ftp-proxy
```

	Total
Current proxy conns	0
Total proxy conns	0
Current Data Proxy	0
Total Data Proxy	0
Total FTP Request	0
Server selection failure	0
no route failure	0
source nat failure	0
feat packet	0
clear ctrl port packet	0
data ssl force	0
request line freed	0
invalid start line	0
auth tls cmd	0
prot cmd	0
pbsz cmd	0
open cmd	0
site cmd	0
user cmd	0
pass cmd	0
quit cmd	0
port cmd	0
cant find port	0
eprt commnad	0

cant find eprt	0
other cmd	0
line too long	0
client auth tls	0
pasv cmd	0
cant find pasv	0
psv addr != svr	0
smp create fail	0
data svr conn fail	0
data send fail	0
epsv command	0
cant find epsv	0
Unsupported auth	0
adat cmd	0
Unsupported PBSZ	0
Unsupported PROT	0
Umsupported cmd	0
Control chn clear txt	0
Control chn ssl	0
Bad Sequence	0
Serv Sel Persist fail	0
Serv Sel SMPv6 fail	0
Serv Sel SMPv4 fail	0
Serv Sel ins tpl fail	0
Client EST state erro	0
Serv CTNG state erro	0
Serv RESP state erro	0
Client RQ state erro	0
Data Start state erro	0
Data Serv CTNG erro	0
Data Serv CTED erro	0
Auth Request	0
Auth Success	0
Auth Failure	0
Forward to Internet	0
Forward to Service-group	0
Drop	0
Resolved Host Name	0
Unresolved Host Name	0

Clearing FTP Proxy Statistics (CLI Procedure)

The `clear slb ftp-proxy` command clears the counters that appear in the `show slb ftp-proxy output`

Displaying FTP Proxy Statistics (GUI Procedure)

To view FTP Proxy statistics, take the following steps:

1. Navigate to **Security >> Forward Proxy >> Statistics** tab.
2. To view the statistics, click **FTP Proxy**.

Clearing FTP Proxy Statistics (GUI Procedure)

To clear the FTP proxy statistics, click **Clear**.

Redirection of Traffic to ICAP Servers

The following topics are covered:

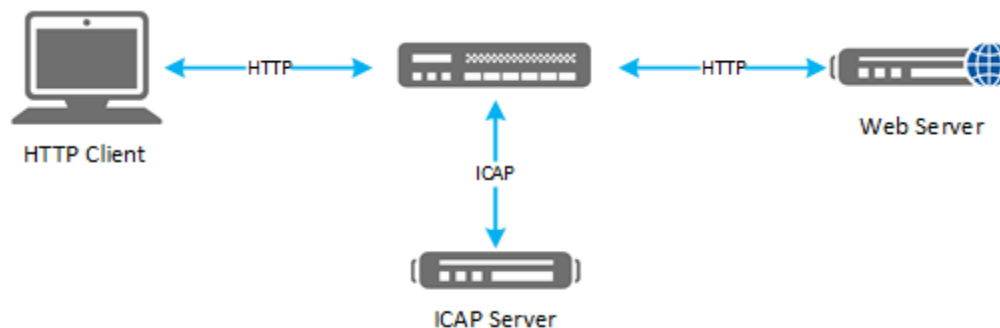
Overview	895
Configuring ICAP	895
ICAP Logging	896

Overview

ACOS supports Internet Content Adaptation Protocol (ICAP) services on HTTP and HTTPS sessions. In other words, ACOS supports configuring ACOS devices to conform to ICAP client recommendations in RFC 3507.

[Figure 86](#) below shows a sample ICAP topology. On traffic from the client to the Web server, ICAP typically serves to provide data loss prevention (DLP). Whereas, on traffic from the Web server to the client, ICAP typically provides anti-virus (AV) services.

Figure 86 : ICAP Support Network Topology



Configuring ICAP

ICAP allows ACOS to integrate with external content adaptation services such as antivirus or content-filtering servers by sending HTTP or HTTPS traffic for inspection and modification.

ICAP can be deployed:

- Independently on standard HTTP or HTTPS ports, or
- Optionally as part of an SSL Insight (SSLi) deployment when encrypted traffic inspection is required.

ICAP Deployment Models

Basic ICAP Deployment

In a basic deployment, ACOS processes HTTP or HTTPS traffic and forwards the content to an external ICAP server for inspection.

This deployment is suitable for environments where:

- Encrypted traffic inspection is not required.
- Encrypted traffic inspection is handled outside ACOS.

The basic ICAP configuration includes:

- Defining the ICAP server
- Creating an ICAP service
- Associating the ICAP service with an HTTP or HTTPS virtual port or policy.

ICAP with SSL Insight Deployment (Optional)

When inspection of encrypted HTTPS traffic is required, ICAP can be deployed with the ACOS SSL Insight (SSLi) feature. In this scenario, ACOS acts as a man-in-the-middle to decrypt SSL traffic before sending content to the ICAP server.

Deployment overviews and configuration instructions for the ACOS man-in-the-middle SSL Insight (SSLi) application are provided in the *SSL Insight Configuration Guide*.

For detailed information on configuring ICAP in an SSLi deployment, see the “ICAP Services” section of the *SSL Insight Configuration Guide*.

ICAP Logging

ACOS provides options to control ICAP logging behavior.

By default, ACOS sends system log entries for all HTTP and HTTPS methods when traffic is forwarded to ICAP servers. These logs record system actions related to forwarding traffic to ICAP services.

You can optionally restrict logging to only specific HTTP methods (such as PUT and POST) that are explicitly defined using the `allowed-http-methods` command. When combined with the `log-only-allowed-method` option, logs are generated only for the specified methods.

This option is supported in both ICAP templates:

- Request Mode (REQMOD)
- Response Mode (RESPMOD)

CLI Configuration

To configure ICAP logging with allowed HTTP methods:

1. Configure a logging server and associate it with a logging service group and template.

```
lb server log 10.1.1.7
    health-check-disable
    port 514 udp
    health-check-disable

slb service-group log udp
    member log 514

slb template logging icap_log
    service-group log
```

2. Configure the ICAP server and create a service group to forward traffic to it.

```
slb server icap 10.1.1.8
    health-check-disable
    port 13440 tcp
    health-check-disable

slb service-group sg-icap13440 tcp
    member icap 13440
```

3. Configure a REQMOD ICAP template, specify the allowed HTTP methods, and

enable logging for only those methods.

```
slb template reqmod-icap reqt
  allowed-http-methods "POST GET PUT HEAD"
  include-protocol-in-uri
  service-url icap://127.0.0.1:13440/example
  service-group sg-icap13440
  template logging icap_log
  log-only-allowed-method
```

4. Configure a RESPMOD ICAP template applying the same logging behavior (Optional).

```
slb template respmod-icap resp
  service-url icap://testuser:1344/resp
  service-group icap
  template logging log-template
  log-only-allowed-method
```

5. Apply the REQMOD ICAP template to an HTTPS virtual server.

```
slb virtual-server vep 10.1.1.100
  port 443 https
  service-group http
  template reqmod-icap reqt
  template client-ssl client-ssl
```

This configuration enables ICAP inspection on HTTPS traffic and ensures that only for the HTTP methods listed in `allowed-http-methods`.

aFlex Configuration

The `log-only-allowed-method` option is also available in aFlex for advanced ICAP filtering. It can be used in combination with the `CATEGORY` command and the ICAP disable logic.

For more information, refer to the *aFlex Scripting Language Reference Document*.

Part IV: ACOS Performance Tuning

This part contains topics pertaining the optimizing the performance of your ACOS device.

The following topics are covered:

- [Stateless Server Load Balancing](#)
- [Stateful Hash-Based Server Load Balancing](#)
- [Auto-Switch to Stateless SLB Based on Traffic](#)
- [TCP-Proxy](#)

Stateless Server Load Balancing

This section describes stateless server load balancing.

The following topics are covered:

Overview	901
Configuration	902

Overview

Stateless SLB conserves system resources by operating without session table entries on the ACOS device. Session table entries contain information about sessions, including client, VIP, and real server IP addresses and protocol ports. Session table entries also may contain additional state information for specific features.

If the ACOS device is running short on sessions, you can use stateless SLB where applicable to make more sessions available for traffic that requires session table entries.

Stateless SLB is valid for the following types of traffic:

- Traffic with very short-lived sessions, such as DNS
- Layer 2 Direct Server Return (DSR) traffic
- Other types of traffic that do not require features that use session-table entries. (See [Stateless Load Balancing Limitations](#).)

You can enable stateless SLB on an individual service-group basis, by selecting a stateless SLB load-balancing method for the group. (See [Configuration](#).)

Methods

Each SLB method defines an algorithm the device uses to distribute traffic among available servers. The following stateless SLB methods are available:

- Stateless Source IP+Port Hash – Balances server load based on a hash value calculated using the source IP address and source TCP or UDP port.
- Stateless Destination IP+Port Hash – Balances server load based on a hash value calculated using the destination IP address and destination TCP or UDP port.
- Stateless Source and Destination IP Hash – Balances server load based on a hash value calculated using both the source and destination IP addresses, and the source and destination TCP or UDP ports.
- Stateless Source IP Only Hash – Balances server load based on a hash value calculated using the source IP address only.

- Stateless Per-Packet Round Robin – Balances server load by sending each packet to a different server, in rotation.
- Stateless Per-Packet Weighted Round Robin – Balances server load based on weight of each server, in rotation. This method is applicable only for traffic that uses a single packet for a request on server level.
- Stateless Per-Packet Service Weighted Round Robin – Balances server load based on weight of each service port, in rotation. This method is applicable only for traffic that uses a single packet for a request on service port level.

Configuration

This section describes configuring stateless load balancing using both GUI and CLI.

GUI Configuration

To use the GUI to configure stateless load balancing:

1. Navigate to **ADC >> SLB >> Service Groups**.
2. Click **Create** or **Edit** in the Actions column for an existing service group.
3. In the Algorithm field, select a stateless load balancing methods (for example, **Stateless SRC DST IP Hash**).
4. Click **Update**.

CLI Configuration

To enable stateless SLB for a service group, use the `method` command at the configuration level for the service group and specify one of the stateless server load balancing algorithms.

The following example configure a stateless SLB service group for UDP traffic. Traffic that is load balanced based on a hash value of the source and destination IP addresses.

```
ACOS(config)# slb service-group dns-stateless udp
ACOS(config-slb svc group)# method stateless-src-dst-ip-hash
ACOS(config-slb svc group)# member dns1 53
ACOS(config-slb svc group-member:53)# exit
```

```
ACOS(config-slb svc group)# member dns2 53  
ACOS(config-slb svc group-member:53)# exit  
ACOS(config-slb svc group)# exit
```

Stateful Hash-Based Server Load Balancing

This section describes stateful hash-based load balancing.

The following topics are covered:

Overview	905
Configuration	905

Overview

Stateful hash-based load balancing provides the persistence and performance benefits of hash-based load balancing, while allowing use of advanced SLB features that require stateful load balancing. For example, rate limiting is an advanced SLB feature that requires stateful load balancing.

One difference between stateless load balancing and stateful load balancing is that stateful load balancing uses the ACOS session table to manage sessions. Stateless load balancing does not use the session table.

Methods

The following stateful hash-based load balancing methods are available:

- src-ip-hash – Hash value based on source IP address and protocol port of client request.
- src-ip-only-hash – Hash value based on only the source IP address of the client request.
- dest-ip-hash – Hash value based on destination IP address and protocol port of client request.
- dest-ip-only-hash – Hash value based on destination IP address of client request.
- odd-even-hash – Two-value hash based on even-odd result of the sum of the source IP address octets.

Configuration

This section describes configuring stateless hash-based load balancing using CLI and GUI.

GUI Configuration

To use the GUI to configure stateful hash-based load balancing:

1. Navigate to **ADC >> SLB >> Service Groups**.
2. Click **Create** or **Edit** in the Actions column for an existing service group.
3. In the Algorithm field, select a stateful load balancing methods (for example, **Source IP Only Hash**).
4. Click **Update**.

CLI Configuration

To use a stateful hash-based load-balancing method, use the `method` command at the configuration level for the service group and specify a stateful load-balancing method. The following commands configure a service group to use stateful hashing based only on source IP address:

```
ACOS(config)# slb service-group stateful-IP-LB tcp
ACOS(config-slb svc group)# method src-ip-only-hash
ACOS(config-slb svc group)# exit
```

Auto-Switch to Stateless SLB Based on Traffic

The following topics are covered:

Overview	908
Configuration	909

Overview

Auto-Switch is an ACOS option that begins dynamically changes the load balancing method for a service group to stateless load balancing based on traffic patterns. The basis of triggering the method change from stateful to stateless load balancing is either exceeding a specified connection rate or Layer 4 session usage.

This feature supports only a single virtual port per service group. If you configure this feature on a service group, you can use that service group with only one virtual port.

Auto-Switch Options

You can configure the following options for this feature. Although only some of these options must be specified when you configure the feature, none of the options has a default value.

- Trigger – The trigger can be one of the following:
 - Connection rate – Rate of new connection requests per second at which the load balancing method is changed. The rate applies collectively to all servers in the service group. The threshold can be 1-1000000 connection requests per second.
 - Layer 4 session use – Percentage of the system-wide Layer 4 session capacity that is currently in use. The threshold can be 1-100 percent.
- Trigger duration – Number of seconds during which the specified trigger must continue to occur before the service group changes to stateless load balancing. Value ranges from 1 to 600 seconds.
- (Optional) Revert trigger – Connection rate or Layer 4 session use percentage at which the service group can revert to using stateful load balancing.
 - For connection rate, the threshold can be 1-1000000 connection requests per second.
 - For Layer 4 session use, the threshold can be 1-100 percent.

- (Optional) Revert duration – Number of seconds when the specified revert trigger must continue to occur before the service group changes to stateful load balancing again. Value ranges from 1 to 600.
- (Optional) Grace period – Interval (seconds) where ACOS device uses current load balancing method for active sessions, before changing to the other load balancing method. Value ranges from 1 to 600.
- The grace period applies only to sessions that are active when the load balancing change is triggered. The change applies immediately to new sessions that begin after the change is triggered.
- Logging – Logs changes between stateful and stateless load balancing that occur due to this feature. Logging is disabled by default.

Configuration

This section describes the configuring auto-switch to stateless SLB using both GUI and CLI.

GUI Configuration

1. Navigate to **ADC >> SLB >> Service Groups** tab.
 - a. If configuring a new service group, select **Create**.
 - b. If modifying an existing service group, click the **Edit** link in the Actions column for the service group.
2. Expand the Advanced Fields category.
3. Select the **Stateless Auto Switch** checkbox to display a drop-down list of available stateless methods.
4. Select the stateless method from the drop-down list. This displays configuration fields for the options described in [Auto-Switch Options](#).
5. To specify the trigger type, select one of the following (See [Auto-Switch Options](#)):
 - a. Connection Rate
 - b. Session Usage

6. Add the servers, if they are not already added.
7. Click **Update**.

CLI Configuration Example 2

These commands configure a service group that uses the stateless-per-pkt-round-robin stateless load-balancing method. This method is used if the rate of new connection requests to the virtual port bound to the service group reaches 80,000 connections per second, and remains at least this high for 300 seconds.

```
ACOS(config)# slb service-group auto-stateless tcp
ACOS(config-slb svc group)# method weighted-rr auto-switch stateless-per-
pkt-round-robin conn-rate 80000 300 60000 300 grace-period 15 log
ACOS(config-slb svc group)# exit
```

To return to using the stateful load-balancing method (weighted round-robin in this example), the rate of new connection requests to the virtual port must drop to 60,000 per second and remain that low for at least 300 seconds. Once this occurs, the ACOS device waits an additional 15 seconds (the grace period) before returning to use of stateful load balancing. Logging is enabled.

To manually reset the service group to stateful SLB, use the `reset auto-switch` command at the configuration level for the service group.

CLI Configuration Example 2

In the following configuration, if Layer 4 session usage reaches 2 percent and stays at least this high for 5 seconds, both service-group members begin using the stateless-dst-ip-hash method. ACOS reverts back to stateful load balancing when 1 percent or less is reached for 5 seconds.

```
slb service-group sg-autol tcp
    method dst-ip-hash auto-switch stateless-dst-ip-hash l4-session-usage
2 5 1 5
    member s1 80
    member s2 80
slb service-group sg-auto tcp
    method dst-ip-hash auto-switch stateless-dst-ip-hash l4-session-usage
2 5 1 5
    member s3 80
    member s4 80
```

TCP-Proxy

The generic TCP-proxy service type provides full TCP-stack service for load-balanced Layer 7 applications. A TCP Proxy server is inserted between a client and destination server, typically to improve some performance aspect. ACOS supports TCP client template and a TCP server template which can each be applied to a TCP-proxy virtual port.

For example:

- A naked ACK option is available for piggybacking protocol/technique in TCP. When this feature is set, the TCP handshake will be complete with a naked ACK (or 'pure' ACK) packet before sending any data.

The following topics are covered:

Configuration	912

Configuration

This section describes configuring generic TCP-proxy.

GUI Configuration

1. Navigate to **ADC >> SLB >> Virtual Servers** tab.
2. Click **Create** to create a new virtual server, or click **Edit** in the Actions column of an existing virtual server.
3. If you are creating a virtual server, you must specify a name and IP address for your new virtual server. If you are editing an existing server, you can skip this step.
4. In the Virtual Port section of the page, click **Create**.
5. In the Protocol field, select **TCP-PROXY** from the drop-down list.
6. Specify the port number.
7. Click **Create**.

Configuring Generic TCP-Proxy (CLI Example)

To assign the TCP-proxy service type to a virtual port, use the `port num tcp-proxy` command at the configuration level for the virtual server.

The following commands configure the real servers, service groups, and VIPs for an IPv4/IPv6 application using the tcp-proxy service type.

These commands configure the real servers:

```
ACOS(config)# slb server s1 10.1.1.3
ACOS(config-real server)# port 80 tcp
ACOS(config-real server-node port)# exit
ACOS(config)# slb server ipv6 fc00::1113
ACOS(config-real server)# port 80 tcp
ACOS(config-real server-node port)# exit
```

These commands configure the service groups:

```
ACOS(config)# slb service-group http tcp
```

```
ACOS(config-slb svc group)# member s1 80  
ACOS(config-slb svc group-member:80)# exit  
ACOS(config-slb svc group)# exit  
ACOS(config)# slb service-group ipv6http tcp  
ACOS(config-slb svc group)# member ipv6 80  
ACOS(config-slb svc group-member:80)# exit  
ACOS(config-slb svc group)# exit
```

These commands configure the VIPs:

```
ACOS(config)# slb virtual-server vip-1 10.153.60.200  
ACOS(config-slb vserver-vport)# exit  
ACOS(config-slb vserver)# exit  
ACOS(config)# slb virtual-server vipipv6 fc00::9000  
ACOS(config-slb vserver-vport)# exit  
ACOS(config-slb vserver)# exit
```

Part V: Protocol Load Balancing

This part contains topics pertaining to protocol load balancing.

The following topics are covered:

- [IPv4 Load Balancing](#)
- [IPv6 Load Balancing](#)
- [Layer 4 TCP/UDP Load Balancing](#)
- [SLB Protocol Translation](#)
- [Proxy Protocol](#)

IPv4 Load Balancing

This section describes load balancing of traffic based solely on transport protocol (TCP, UDP, or others such as ICMP), without the need to specify the protocol port numbers to be load balanced.

The following topics are covered:

Overview	916
Configuring IPv4 Load Balancing	919

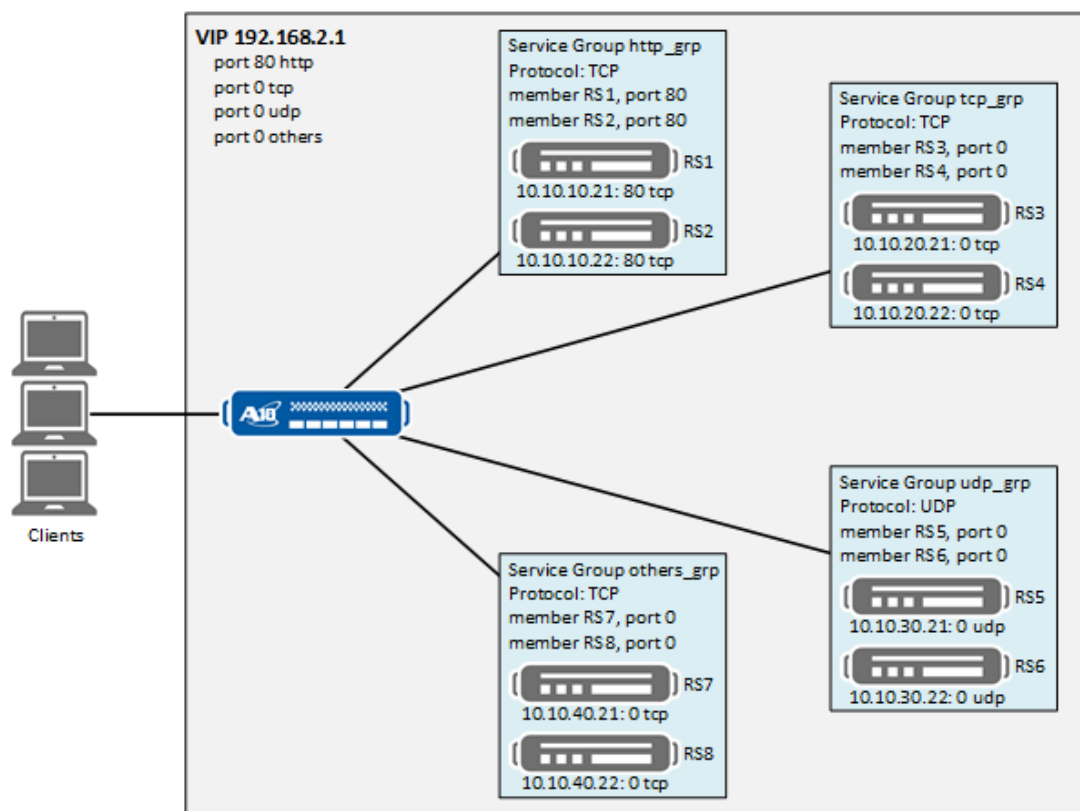
Overview

IP protocol load balancing enables you to easily load balance traffic based solely on whether the traffic is TCP, UDP, or others such as ICMP (not UDP or TCP), without the need to specify the protocol port numbers to be load balanced.

You can combine IP protocol load balancing with other load balancing configurations. For example, you can use IP protocol load balancing along with HTTP load balancing. In this case, HTTP traffic to the VIP HTTP port number is load balanced separately from traffic to other port numbers. [Figure 87](#) shows a hypothetical example of an IP protocol load balancing deployment.

NOTE: For a real-world example, see *A10 Microsoft Exchange Server Deployment Guide* on A10 Networks, Inc. website.

Figure 87 : IP Protocol Load Balancing



This example uses separate service groups for each of the following types of traffic:

- HTTP traffic addressed to TCP port 80 is sent to service group http-grp.
- All TCP traffic addressed to any TCP port except port 80 is sent to service group tcp-grp.
- All UDP traffic, addressed to any UDP port, is sent to service group udp-grp.
- All other traffic (all non TCP/UDP traffic) is sent to service group others-grp.

Although this example shows separate service groups for each type of traffic, you can use the same service group for multiple traffic types.

In IP protocol load-balancing configurations, port 0 (zero) is used as a wildcard port and matches on any port number. In configurations where some protocol port numbers are explicitly specified, SLB for those ports takes precedence over SLB for

the wildcard port (0). In the example above, the service group configured for TCP port 80 is always used for client requests addressed to that port, instead of a service group configured for the wildcard port.

Health checking does not apply to the wildcard port. When you configure IP protocol load balancing, make sure to disable health checking of port 0. If you leave health checking enabled, the port will be marked down and the client's request therefore will not be serviced.

Comparison of IP Protocol Load Balancing to Layer 4 TCP/UDP Load Balancing

IP protocol load balancing is similar to Layer 4 load balancing, except IP protocol load balancing enables you to load balance non-TCP/UDP traffic. Layer 4 load balancing applies only to TCP or UDP traffic. In addition, IP protocol load balancing uses a wildcard port number that matches on any TCP port, UDP port, or any non-TCP/UDP port, depending on the configuration. Layer 4 load balancing requires you to explicitly specify the protocol port numbers to load balance.

NAT Support

For client request traffic to which IP protocol load balancing applies, the ACOS device translates only the destination IP address, not the protocol port number. ACOS translates the destination IP address in the request from the VIP address to a real server's IP address. ACOS then sends the request to the same protocol port number as the one requested by the client. (Port is translated unless port 0 is used.)

In configurations where some protocol port numbers are explicitly specified, auto port translation is still supported for the explicitly specified port numbers. In the example above, NAT can translate TCP port 80 into another TCP port number if required by the configuration.

Template Support

- For TCP or UDP, a TCP or UDP template is applied, as in other types of SLB. Optionally, you can also use a source-IP persistence template.
- For non-TCP/UDP traffic, the TCP template is used.

Direct Server Return

For either of the following types of applications, IP protocol load balancing is supported only when Direct Server Return (DSR) is enabled on the virtual port.

- Application Layer Gateway (ALG) applications, such as FTP. For an ALG application, either enable DSR or configure SLB explicitly for the ALG service port.
- Any application that requires inspection of any part of the client request packet other than the destination IP address

NOTE: In the CLI, DSR is enabled by the `no-dest-nat` command.

Configuring IPv4 Load Balancing

To configure IP protocol load balancing:

1. Configure the real servers. For each real server that will service requests to IP protocol load-balanced traffic, add service port 0 (the wildcard port).
Disable health checking of port 0. Health checking does not apply to the wildcard port.
2. Configure the service group(s). To add members (real servers) for traffic to which IP protocol load balancing will apply, specify 0 as the protocol port for the member.
3. Configure the virtual server. Bind virtual port 0 to the service group(s) that have members. Specify one of the following as the service type:
 - a. TCP
 - b. UDP
 - c. Others

NOTE: For load balancing of non-TCP/UDP traffic such as ICMP, you can specify TCP or UDP as the transport protocol, in the configurations of the real server ports and service groups. If the port number is 0 and the service type on the virtual port is “others”, the ACOS device will load balance the traffic as non-TCP/UDP traffic.

GUI Configuration

Configuration of IP protocol SLB is similar to configuration of TCP/UDP SLB, with the following differences.

1. Real Server Port section: Navigate to **ADC >> SLB >> Servers >> Port**, enter 0 in the **Port** field.
2. Service Groups section: Navigate to **ADC >> SLB >> Service Groups >> Member >> Port**, enter 0 as the port number.
3. Virtual Port section: Navigate to **ADC >> SLB >> Virtual Servers >> Virtual Port**, select the protocol from the **Protocol** drop-down list.

CLI Configuration

The following commands configure the real servers shown in [IP Protocol Load Balancing](#).

For simplicity, the example assumes that only the default TCP health check is used for port 80. Health checking does not apply to the wildcard port number and is therefore disabled. Health checking of other, explicitly specified port numbers is still supported as in previous releases.

```
ACOS(config)# slb server RS1 10.10.10.21
ACOS(config-real server)# port 80 tcp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
ACOS(config)# slb server RS2 10.10.10.22
ACOS(config-real server)# port 80 tcp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
ACOS(config)# slb server RS3 10.10.20.21
ACOS(config-real server)# port 0 tcp
ACOS(config-real server-node port)# health-check-disable
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
ACOS(config)# slb server RS4 10.10.20.22
ACOS(config-real server)# port 0 tcp
ACOS(config-real server-node port)# health-check-disable
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
```

```
ACOS(config)# slb server RS5 10.10.30.21
ACOS(config-real server)# port 0 udp
ACOS(config-real server-node port)# health-check-disable
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
ACOS(config)# slb server RS6 10.10.30.22
ACOS(config-real server)# port 0 udp
ACOS(config-real server-node port)# health-check-disable
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
ACOS(config)# slb server RS7 10.10.40.21
ACOS(config-real server)# port 0 tcp
ACOS(config-real server-node port)# health-check-disable
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
ACOS(config)# slb server RS8 10.10.40.22
ACOS(config-real server)# port 0 tcp
ACOS(config-real server-node port)# health-check-disable
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
```

The following commands configure the service groups.

```
ACOS(config)# slb service-group http-grp tcp
ACOS(config-slb svc group)# member RS1 80
ACOS(config-slb svc group-member:80)# exit
ACOS(config-slb svc group)# member RS2 80
ACOS(config-slb svc group-member:80)# exit
ACOS(config-slb svc group)# exit
ACOS(config)# slb service-group tcp-grp tcp
ACOS(config-slb svc group)# member RS3 0
ACOS(config-slb svc group-member:0)# exit
ACOS(config-slb svc group)# member RS4 0
ACOS(config-slb svc group-member:0)# exit
ACOS(config-slb svc group)# exit
ACOS(config)# slb service-group udp-grp udp
ACOS(config-slb svc group)# member RS5 0
ACOS(config-slb svc group-member:0)# exit
ACOS(config-slb svc group)# member RS6 0
ACOS(config-slb svc group-member:0)# exit
ACOS(config-slb svc group)# exit
```

```
ACOS(config)# slb service-group others-grp tcp
ACOS(config-slb svc group)# member RS7 0
ACOS(config-slb svc group-member:0)# exit
ACOS(config-slb svc group)# member RS8 0
ACOS(config-slb svc group-member:0)# exit
ACOS(config-slb svc group)# exit
```

The following commands configure the virtual server.

```
ACOS(config)# slb virtual-server vip1 192.168.2.1
ACOS(config-slb vserver)# port 80 tcp
ACOS(config-slb vserver-vport)# service-group http-grp
ACOS(config-slb vserver-vport)# exit
ACOS(config-slb vserver)# port 0 tcp
ACOS(config-slb vserver-vport)# service-group tcp-grp
ACOS(config-slb vserver-vport)# exit
ACOS(config-slb vserver)# port 0 udp
ACOS(config-slb vserver-vport)# service-group udp-grp
ACOS(config-slb vserver-vport)# exit
ACOS(config-slb vserver)# port 0 others
ACOS(config-slb vserver-vport)# service-group others-grp
ACOS(config-slb vserver-vport)# exit
ACOS(config-slb vserver)# exit
```

To display configuration information and statistics, use the show commands used for other types of SLB:

```
ACOS(config)# show slb virtual
ACOS(config)# show slb server
ACOS(config)# show slb service-group
ACOS(config)# show session
```

IPv6 Load Balancing

The following topics are covered:

Overview	924
CLI Configuration	924

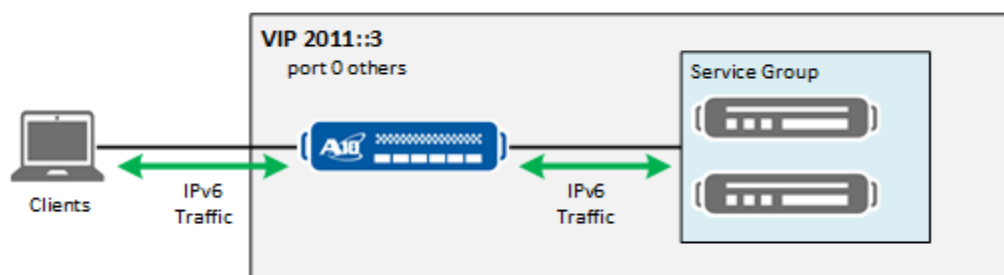
Overview

This section describes how to configure IPv6 IP load balancing on virtual servers and wildcard VIPs. IPv6 IP load balancing for ICMP and tunnel protocols such as IPIP6 also is supported.

IPv4 to IPv6 and IPv6 to IPv4 port wildcard load balancing is not supported in the current release.

Support for IPv6 IP load balancing is end to end as shown in the following graphic:

Figure 88 : IPv6 Load Balancing End to End



CLI Configuration

IPv6 Load Balancing Configuration (CLI Example 1)

For IPv6 load balancing with a regular VIP (including an ICMP echo request/reply), follow these steps:

1. On the ACOS device, use the following commands (configuration commands for the v6tcp service group are not included):

```
ACOS(config)# slb virtual-server v6 2011::3
ACOS(config-slb vserver)# port 0 others
ACOS(config-slb vserver-vport)# name _v6_2011_3_others_65535
ACOS(config-slb vserver-vport)# service-group v6tcp
```

2. On a client, issue the `ping ipv6 2011::3` command.

3. On the ACOS device, issue a **show session** command. You may repeat the steps 2 and 3 on multiple clients using the same ACOS VIP.
4. With the configuration on the ACOS device, the **ping** command will function normally and an ICMP session will be created on the ACOS device.

IPv6 Load Balancing Configuration (CLI Example 2)

For IPv6 load balancing with a regular VIP, and with an IPIP6 tunnel configured on both the client and the server, use the following commands:

1. On the ACOS device, issue the following commands (configuration commands for the v6tcp service group are not included):

```
ACOS(config)# slb virtual-server v6 2011::3
ACOS(config-slb vserver)# port 0 others
ACOS(config-slb vserver-vport)# name _v6_2011_3_others_65535
ACOS(config-slb vserver-vport)# service-group v6tcp
```

2. On a client and a server, configure an ipip6 tunnel.
3. Run traffic on this tunnel.
4. On the ACOS, issue a **show session** command. You may repeat the steps 2 and 3 on multiple clients using the same ACOS VIP.

With this configuration, tunnel traffic works correctly with an IP session created on the ACOS device.

IPv6 Load Balancing Configuration (CLI Example 3)

For IPv6 load balancing with a wildcard VIP and an ICMP echo request/reply, running-config looks like this:

```
slb virtual-server wv6 ::
  port 0  others
  name _wildcard_v6_v6_Others_0
  service-group gw
  no-dest-nat
```

With this configuration on the ACOS device, the ping command will function normally and an ICMP session will be created on the ACOS device.

1. Configure device using the following configuration steps (configuration commands for the gw service group are not included):

```
ACOS(config)# slb virtual-server wv6 ::
ACOS(config-slb vserver)# port 0 others
ACOS(config-slb vserver-vport)# name _wildcard_v6_v6_Others_0
ACOS(config-slb vserver-vport)# service-group gw
```

2. On the client, execute `ping ipv6 2011::10`.
3. On the ACOS device, execute the `show session` command.
4. Repeat step 2 and 3 from multiple clients on the same ACOS VIP.
5. On another client, issue the `ping ipv6 2011::11` command.
6. On the ACOS device, execute CLI `show session` command.

IPv6 Load Balancing Configuration (CLI Example 4)

For IPv6 load balancing with a wildcard VIP in a L3V partition, and with an IPIP6 tunnel configured on both the client and the server, your running configuration will look like the following:

```
slb virtual-server wv6 ::
  port 0  others
    name _wildcard_v6_v6_Others_0
    service-group gw
    no-dest-nat
```

With this configuration, traffic on the tunnel works correctly with an IP session created on the ACOS device.

1. Configure the ACOS device using the following configuration commands (configuration commands for the gw service group are not included):

```
ACOS(config)# slb virtual-server wv6 ::
ACOS(config-slb vserver)# port 0 others
ACOS(config-slb vserver-vport)# name _wildcard_v6_v6_Others_0
ACOS(config-slb vserver-vport)# service-group gw
ACOS(config-slb vserver-vport)# no-dest-nat
```

2. On the client and server, configure the ipip6 tunnel.
3. Run traffic on this tunnel.

4. On the ACOS device, execute the `show session` command.
5. While the session with the current partition exists, repeat steps 2, 3, and 4 in a different partition.

Layer 4 TCP/UDP Load Balancing

The following topics are covered:

Overview	929
Configuring Layer 4 Load Balancing	932
TCP Fast Open	937

Overview

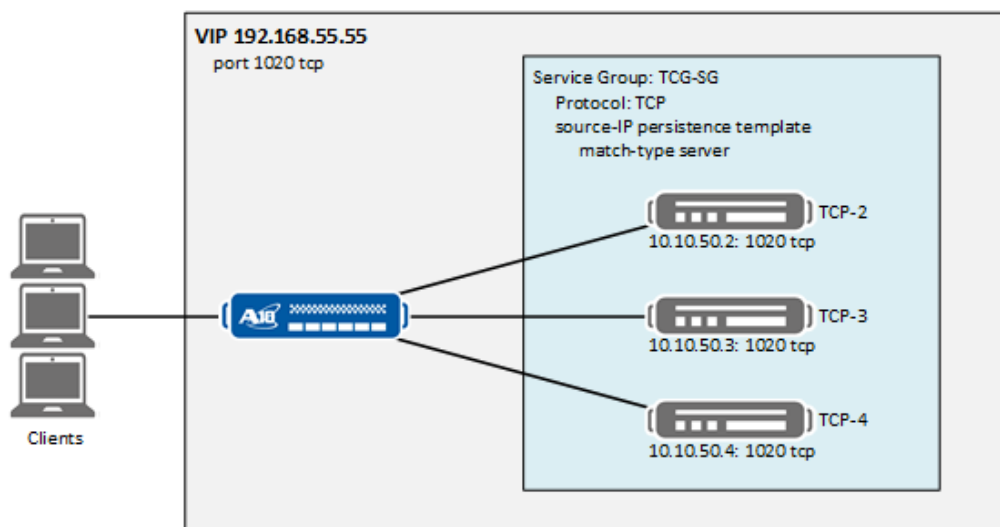
This section describes Layer 4 load balancing of TCP and UDP traffic and how to configure it.

Layer 4 load balancing requires specifying the protocol port numbers to be load balanced. To load balance traffic based solely on transport protocol (TCP, UDP, or other), see [IPv4 Load Balancing](#).

In addition to load balancing for well-known and widely used types of services such as HTTP, HTTPS, and FTP, ACOS devices also support Layer 4 load balancing for custom applications. If a service you need to load balance is not one of the well-known service types recognized by the ACOS device, you still can configure Layer 4 TCP or UDP load balancing for the service.

[Figure 89](#) shows an example of a Layer 4 load balancing implementation.

Figure 89 : Layer 4 SLB



Layer 4 load balancing balances traffic based on the transport protocol (TCP or UDP) and the protocol port number. The payload of the UDP or TCP packets is not examined.

In this example, a custom application is running on a server farm consisting of three real servers. Clients navigate to the VIP to use the custom application.

Layer 4 Load Balancing Data Structures

This section describes the following data structures that configure Layer 4 load balancing.

The following topics are covered:

Service Groups	930
Virtual Server	930
Templates	930
Idle Timeouts	930

Service Groups

This example uses a single service group that contains all the real servers. The service group uses the default load balancing method (round robin).

Virtual Server

The custom application on the real servers is accessed at TCP port 1020 by clients through virtual IP address 192.168.55.55.

Templates

ACOS has default TCP and UDP templates. You can use the default template or configure another TCP or UDP template and use that one instead. If your Layer 4 load balancing configuration is for a TCP application and you do not bind a TCP template to the virtual port, the default TCP template is used. For a UDP application, the default UDP template is used unless you bind another UDP template to the virtual port.

Idle Timeouts

One of the parameters you can configure in TCP and UDP templates is the idle time. Depending on the requirements of your application, you can reduce or increase the

amount of time the ACOS device allows a session to remain idle.

For UDP transaction-based applications, another parameter you can adjust is how quickly connections are terminated after a server reply is received. For example, if there are licensing costs associated with active sessions, you can minimize unnecessary costs by quickly terminating idle sessions, and immediately terminating connections that are no longer needed.

- For more information about TCP template parameters, see the `slb template tcp` command in the *Command Line Interface Reference*.
- For more information about UDP template parameters, see the `slb template udp` command in the *Command Line Interface Reference*.

Configurable TCP Half-Open Timer

This feature is a configurable TCP half-open timeout that is independent of TCP idle-timeout. The default timer for half-open connections is was 60 seconds. Configurable half-open timeout values range between 1 and 60 seconds in one-second increments. Half-open connections are visible in the `show session` command output.

GUI Configuration

The menu path to the TCP half-open timeout option is **ADC >> Templates >> L4 Protocols >> Create >> TCP >> *Template Name***. You can access the option both in a configured TCP or TCP-Proxy template.

CLI Configuration

The `half-open-idle-timeout` command enables aging of half-open TCP sessions. A half-open TCP session is one in which the client receives a SYN-ACK, but does not reply with an ACK. You can set the timeout value to 1-60 seconds. The default value is 60.

Source-IP Persistence

Optionally, you also can configure a source-IP persistence template and bind it to the virtual port. The example in this section uses a source-IP persistence template that is configured to send all traffic from a given client IP address to the same real server.

Without this custom template, different requests from a given client can be sent to different servers, based simply on the load balancing method.

For more information about the source-IP persistence template parameters, see the `slb template persist source-ip` command in the *Command Line Interface Reference*.

GUI Configuration

1. Navigate to **ADC >> Templates >> L4 Protocols** tab.
2. Click **Create**.
3. Select TCP or UDP. The Create TCP Template window appears.
4. Enter a **Name** for the template.
5. Edit template settings as needed for your application.
6. Click **OK**.

Layer 4 Load Balancing Health Monitors

The example uses default Layer 3 and Layer 4 health monitors. The Layer 3 monitor (Ping) and applicable Layer 4 monitor (TCP or UDP) are enabled by default when you configure real server and real service ports.

You can create an external health monitor using a script and import the monitor onto the ACOS device. For information, see [Health Monitoring](#).

Configuring Layer 4 Load Balancing

Configuring Layer 4 Load Balancing (Procedure)

To configure Layer 4 load balancing:

1. Configure the real servers. Add the custom application's TCP or UDP port number, with the applicable service type (TCP or UDP).
2. Configure a service group. Add the real servers, service port, and any custom templates to the group.
3. If applicable, configure a custom TCP or UDP template.

4. If applicable, configure a source-IP persistence template.
5. Configure the virtual server. Bind the virtual service port on the virtual server to the service group and custom templates, if configured.

GUI Configuration

The example consists of the following steps:

- [Configuring the Real Servers](#)
- [Configuring the service group](#)
- [Configuring a source-IP persistence template](#)
- [Configuring the virtual server](#)

Configuring the Real Servers

1. Navigate to **ADC >> SLB >> Servers** tab.
2. Click **Create**.
3. Configure basic settings for the server, such as **Name** and IP address **Type** (IPv4 or IPv6).
4. In the Port section, click **Create**.
5. In the **Port Number** field, enter the protocol port number for the application.
6. Click the **Protocol** drop-down menu and select the transport protocol for the application, **TCP** or **UDP**.
7. Configure other required port settings, then click **Create**. The application port appears in the Port list.
8. Click **Create** (or Update, if modifying an existing server). The real server appears in the real server table.

Configuring the service group

1. Navigate to **ADC >> SLB >> Service Groups** tab.
2. Click **Create**.
3. Enter a **Name** for the service group.

4. Click the **Protocol** drop-down menu and select the transport protocol for the application, **TCP** or **UDP**.
5. Click **Create**.
6. In the Member section of the window, click the Create button. The Create Member page appears.
7. Select the **Choose Creation Type** radio button:
 - a. **Existing Server** – if you wish to modify an existing server.
 - b. **New Server** – if you wish to create a new server for this service group (requires entering name and IP).
8. Enter the protocol port number in the **Port** field.
9. Click **Create**.
10. Repeat step 7 through step 10 for each server and port.
11. Click **Update**. The service group appears in the Service Groups table.

Configuring a source-IP persistence template

1. Navigate to **ADC >> Templates >> Persistence** tab.
2. Click **Create** and from the drop-down menu that appears, select **Persist Source IP**.
3. Enter a name for the template.
4. Edit template settings as needed for your application.
5. Click **OK**.

Configuring the virtual server

1. Navigate to **ADC >> SLB >> Virtual Servers** tab.
2. Click **Create**. The Create Virtual Server page appears.
3. Enter a name for the virtual server.
4. For the **Address Type** radio button, select either IPv4 or IPv6.
5. In the **IP Address** field, enter the virtual IP address to which clients will send requests.
6. Configure other general settings as needed for your deployment.

7. In the Virtual Port section, click **Create**. The Create Virtual Port window appears.
8. Enter a name for the virtual port in the **Name** field.
9. In the **Protocol** drop-down list, select the transport protocol for the application, TCP or UDP.
10. Enter the application port number in the **Port** field.
11. If you configured any custom templates, select them from the drop-down lists for each template type.
12. Enter or select other values as needed.
13. Click **Create**. The new virtual port appears in the table.
14. Click **Update**. The virtual server appears in the virtual server table.

CLI Configuration

The example configures Layer 4 Load Balancing:

- The following commands configure the real servers:

```
ACOS(config)# slb server tcp-2 10.10.50.2
ACOS(config-real server)# port 1020 tcp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
ACOS(config)# slb server tcp-3 10.10.50.3
ACOS(config-real server)# port 1020 tcp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
ACOS(config)# slb server tcp-4 10.10.50.4
ACOS(config-real server)# port 1020 tcp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
```

- The following commands configure the service group “tcp-sg” and adds the real servers as members:

```
ACOS(config)# slb service-group tcp-sg tcp
ACOS(config-slb svc group)# member tcp-2 1020
ACOS(config-slb svc group-member:1020)# exit
ACOS(config-slb svc group)# member tcp-3 1020
ACOS(config-slb svc group-member:1020)# exit
```

```
ACOS(config-slb svc group)# member tcp-4 1020
ACOS(config-slb svc group-member:1020)# exit
ACOS(config-slb svc group)# exit
```

- The following commands configure a source-IP persistence template:

```
ACOS(config)# slb template persist source-ip app1020persist
ACOS(config-source ip persistence template)# match-type server
ACOS(config-source ip persistence template)# exit
```

- The following commands configure the virtual server:

```
ACOS(config)# slb virtual-server web-vip 192.168.55.55
ACOS(config-slb vserver)# port 1020 tcp
ACOS(config-slb vserver-vport)# service-group tcp-sg
ACOS(config-slb vserver-vport)# template persist source-ip app1020persist
ACOS(config-slb vserver-vport)# exit
ACOS(config-slb vserver)# exit
```

Configuring Radius UDP Template by Disabling Persist Scoring

1. Persist scoring is a default method used for internal server selection. Disable the default global persist scoring by using the following command.

```
ACOS(config)# slb common
ACOS(config-common)# disable-persist-scoring
```

2. The persist scoring can be overridden at the service group level by enabling or disabling persist scoring for the service group.

```
ACOS(config)# slb common
ACOS(config-common)# persist-scoring enable
```

3. Set IPv4 offset for the load balancing hash. The offset is a mask that masks certain octets of the source IP. The next command sets offset of 2:

```
ACOS(config)# slb common
ACOS(config-common)# lb-method-hash src-ip-only ipv4-offset 2
```

4. Configure the RADIUS template with the AVP (Attribute Value Pair) value. The AVP number is a Radius attribute (8=Framed IP and 4= NAS-IP) present in the RADIUS packet. :

```
ACOS(config)# slb template udp udp-radius
ACOS(config)# radius-lb-method hash ip avp 8
```

For more information on these commands, see *Command Reference Guide*.

TCP Fast Open

TCP Fast Open Overview

TCP Fast Open (TFO) is an update to TCP that enables data to be exchanged safely and save one full round-trip time (RTT) over the standard three-way handshake during a TCP session. TFO allows data in SYN packets to be delivered to the application unlike standard TCP and thus yields significant latency improvement. The main component of TFO is the security cookie, which is used by the server to authenticate the client initiating the TCP connection.

The following list outlines how the client requests a TFO cookie:

1. The client sends a SYN with data and the security cookie.
2. The server validates the cookie:
 - a. If the cookie is valid, the server sends a SYN-ACK acknowledging both the SYN and the data. The server then delivers the data to the application.
 - b. Otherwise, the server drops the data and sends a SYN-ACK acknowledging only the SYN sequence number.
3. If the server accepts the data in the SYN packet, it sends the response during the handshake. The data limit is governed by TCP congestion control.
4. The client then sends an ACK acknowledging the server data, retransmits the data in the ACK packet until it is acknowledged by the server.

The rest of the connection proceeds like a normal TCP connection. The client can repeat many TFO operations once it acquires a cookie (until the cookie is expired by the server). Thus, TFO is useful for applications that have temporal locality on client and server connections.

Configuring TCP Fast Open

The following parameters can be configured for TCP stack Fast Open after creating the virtual server template:

slb template virtual-server

Example: ACOS(config)# slb template virtual-server default

Table 22 : SLB Template Virtual Server Parameters

Parameter	Description	Example
tcp-stack-tfo-active-conn-limit	Sets the limit for active layer 7 tcp fast open connection. Default is zero.	ACOS(config-vserver)# tcp-stack-tfo-active-conn-limit 7
tcp-stack-tfo-backoff-time	Sets the time(in seconds) for which tcp stack waits before allowing new fast open requests after security condition. Default is 600 seconds	ACOS(config-vserver)# tcp-stack-tfo-backoff-time 700
tcp-stack-tfo-cookie-time-limit	Sets the time (in seconds) for which a layer 7 tcp fast open cookie is valid. Default is 60 seconds	ACOS(config-vserver)# tcp-stack-tfo-cookie-time-limit 70

SLB Protocol Translation

The following topics are covered:

Overview	940
Requirement	941
Configuring SLB Protocol Translation	941

Overview

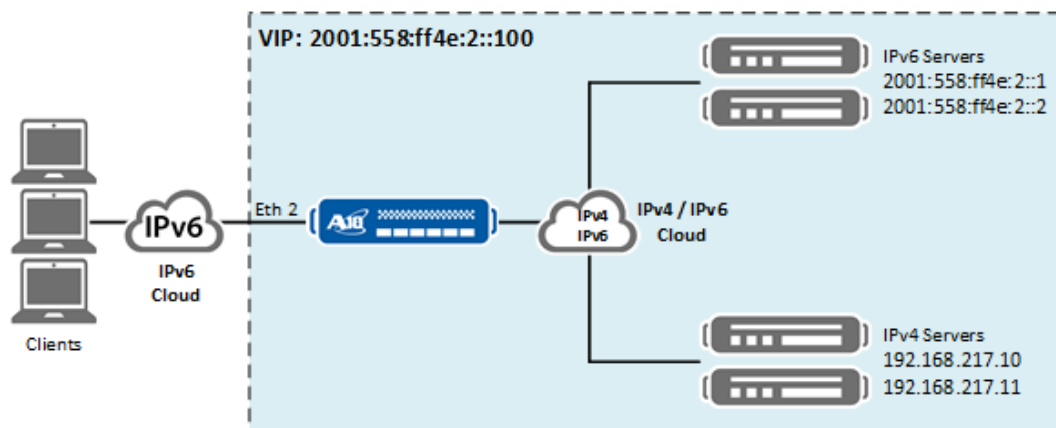
SLB Protocol Translation (SLB-PT) enables IPv4 servers to be used for serving content to IPv6 clients. Likewise, SLB-PT enables IPv6 servers to be used for serving content to IPv4 clients. Server farms can contain both IPv4 and IPv6 servers.

SLB-PT is supported for the following virtual port types:

- FTP
- UDP
- TCP
- HTTP
- HTTPS
- SSL-proxy
- SMTP

[SLB Protocol Translation](#) shows an example of a SLB-PT deployment that uses a mixed server farm of IPv4 and IPv6 servers to serve IPv6 clients.

Figure 90 : SLB Protocol Translation



In this example, a server farm consisting of IPv6 and IPv4 servers is configured with an IPv6 VIP address. IPv6 clients send requests to the IPv6 VIP. ACOS then selects an

IPv6 or IPv4 server and forwards the client's request to the selected server. If the server is an IPv4 server, the ACOS device translates the IP protocol of the client's request from IPv6 to IPv4 before forwarding it to the IPv4 server. Likewise, when the ACOS device receives the server's reply, the ACOS device translates the reply from IPv4 to IPv6, then forwards the reply to the client.

Requirement

In addition to the standard SLB configuration items (servers, service groups, the VIP, and so on), SLB-PT requires IP source NAT.

As a minimum requirement, a single NAT pool is required, for the IP type (IPv4 or IPv6) that differs from the IP type of clients. In this example, an IPv4 pool is required. The pool is used if the ACOS device selects an IPv4 server for an IPv6 client's request. The pool must be bound to each of the virtual ports that has a corresponding real port on an IPv4 server.

If the deployment also will send IPv4 client requests to IPv6 servers, an IPv6 pool is also required.

For simplicity, the CLI example below uses a single IPv4 NAT pool. Following the example, the [Multiple NAT Pools - CLI Configuration](#) section describes how to use multiple pools.

Configuring SLB Protocol Translation

This section includes the following SLB Protocol Translation examples:

- [Single NAT Pool - CLI Configuration](#)
- [Multiple NAT Pools - CLI Configuration](#)

Single NAT Pool - CLI Configuration

The following commands configure the SLB-PT deployment shown in [SLB Protocol Translation](#).

The following commands configure the Ethernet interfaces connected to the clients and servers:

```
ACOS(config)# interface ethernet 1
ACOS(config-if:ethernet:1)# ip address 192.168.217.100 255.255.255.0
ACOS(config-if:ethernet:1)# ipv6 address fc00:558:ff4e:2::100/64
ACOS(config-if:ethernet:1)# enable
ACOS(config-if:ethernet:1)# exit
ACOS(config)# interface ethernet 2
ACOS(config-if:ethernet:2)# ipv6 address fc00:32::2020:2001/64
ACOS(config-if:ethernet:2)# enable
ACOS(config-if:ethernet:2)# exit
```

The following command configures an IPv4 source NAT pool.

```
ACOS(config)# ip nat pool v4natpool-1 192.168.217.200 192.168.217.202
netmask /24
```

For simplicity, this example uses only a single pool. If multiple pools are used, ACLs are also required. The ACLs must match on the client IP address(es) as the source address. If the real servers and VIP are in different subnets, the ACLs also must match on the real server IP address(es) as the destination address. (For more information, see [Multiple NAT Pools - CLI Configuration](#). Also see the “Network Address Translation” section in the *System Configuration and Administration Guide*.)

These commands configure the IPv4 real servers. The IPv4 and IPv6 servers use the same real ports.

```
ACOS(config)# slb server v4server-1 fc00:558:ff4e:2::1
ACOS(config-real server)# port 80 tcp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# port 53 udp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# port 443 tcp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# port 25 tcp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
ACOS(config)# slb server v4server-2 fc00:558:ff4e:2::2
ACOS(config-real server)# port 80 tcp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# port 53 udp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# port 443 tcp
ACOS(config-real server-node port)# exit
```

```
ACOS(config-real server)# port 25 tcp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
```

The following commands configure the IPv6 real servers:

```
ACOS(config)# slb server v6server-1 fc00:558:ff4e:2::1
ACOS(config-real server)# port 80 tcp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# port 53 udp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# port 443 tcp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# port 25 tcp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
ACOS(config)# slb server v6server-2 fc00:558:ff4e:2::2
ACOS(config-real server)# port 80 tcp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# port 53 udp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# port 443 tcp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# port 25 tcp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
```

The following commands configure the service groups. A separate service group is configured for each application (real port):

```
ACOS(config)# slb service-group sg4v6-http tcp
ACOS(config-slb svc group)# member v4server-1 80
ACOS(config-slb svc group-member:80)# exit
ACOS(config-slb svc group)# member v4server-2 80
ACOS(config-slb svc group-member:80)# exit
ACOS(config-slb svc group)# member v6server-1 80
ACOS(config-slb svc group-member:80)# exit
ACOS(config-slb svc group)# member v6server-2 80
ACOS(config-slb svc group-member:80)# exit
ACOS(config-slb svc group)# exit
ACOS(config)# slb service-group sg4v6-dns udp
ACOS(config-slb svc group)# member v4server-1 53
```

```

ACOS(config-slb svc group-member:53)# exit
ACOS(config-slb svc group)# member v4server-2 53
ACOS(config-slb svc group-member:53)# exit
ACOS(config-slb svc group)# member v6server-1 53
ACOS(config-slb svc group-member:53)# exit
ACOS(config-slb svc group)# member v6server-2 53
ACOS(config-slb svc group-member:53)# exit
ACOS(config-slb svc group)# exit
ACOS(config)# slb service-group sgw4v6-https tcp
ACOS(config-slb svc group)# member v4server-1 443
ACOS(config-slb svc group-member:443)# exit
ACOS(config-slb svc group)# member v4server-2 443
ACOS(config-slb svc group-member:443)# exit
ACOS(config-slb svc group)# member v6server-1 443
ACOS(config-slb svc group-member:443)# exit
ACOS(config-slb svc group)# member v6server-2 443
ACOS(config-slb svc group-member:443)# exit
ACOS(config-slb svc group)# exit
ACOS(config)# slb service-group sgw4v6-smtp tcp
ACOS(config-slb svc group)# member v4server-1 25
ACOS(config-slb svc group-member:25)# exit
ACOS(config-slb svc group)# member v4server-2 25
ACOS(config-slb svc group-member:25)# exit
ACOS(config-slb svc group)# member v6server-1 25
ACOS(config-slb svc group-member:25)# exit
ACOS(config-slb svc group)# member v6server-2 25
ACOS(config-slb svc group-member:25)# exit
ACOS(config-slb svc group)# exit
ACOS(config)# exit

```

The following commands import an SSL certificate and key, and configure a client-SSL template to use them. ACOS will use the certificate and key to terminate SSL sessions between clients and the VIP.

```

ACOS# import cert sslcert.pem scp:
Address or name of remote host []?10.10.10.2
User name []?ACOSadmin
Password []?*****
File name [/]?sslcert.pem
ACOS# import key certkey.pem scp:
Address or name of remote host []?10.10.10.2

```

```

User name []?ACOSadmin
Password []?*****
File name [/]?certkey.pem
ACOS# config
ACOS(config)# slb template client-ssl cssl
ACOS(config-client SSL template)# certificate sslcert.pem key certkey.pem
ACOS(config-client SSL template)# exit

```

The following commands configure the VIP:

```

ACOS(config)# slb virtual-server v6vip FD00:32::2020:2000
ACOS(config-slb vserver)# port 80 http
ACOS(config-slb vserver-vport)# source-nat pool v4natpool-1
ACOS(config-slb vserver-vport)# service-group sgv4v6-http
ACOS(config-slb vserver-vport)# exit
ACOS(config-slb vserver)# port 53 udp
ACOS(config-slb vserver-vport)# source-nat pool v4natpool-1
ACOS(config-slb vserver-vport)# service-group sgv4v6-dns
ACOS(config-slb vserver-vport)# exit
ACOS(config-slb vserver)# port 443 https
ACOS(config-slb vserver-vport)# source-nat pool v4natpool-1
ACOS(config-slb vserver-vport)# template client-ssl cssl
ACOS(config-slb vserver-vport)# service-group sgv4v6-https
ACOS(config-slb vserver-vport)# exit
ACOS(config-slb vserver)# port 25 smtp
ACOS(config-slb vserver-vport)# source-nat pool v4natpool-1
ACOS(config-slb vserver-vport)# service-group sgv4v6-smtp
ACOS(config-slb vserver-vport)# exit
ACOS(config-slb vserver)# exit

```

Multiple NAT Pools - CLI Configuration

The example shown above uses only a single NAT pool, for access to the IPv4 servers. If multiple pools are used, then different CLI syntax is required.

Multiple IPv4 Pools

Here is an example that uses multiple IPv4 pools.

First, IPv6 ACLs that match on the client IP address(es) are configured. A separate ACL is required for each NAT pool.

```
ACOS(config)# ipv6 access-list v6acl-1
ACOS(config-access-list:v6acl-1)# permit ipv6 fc00:32::/96 any
ACOS(config-access-list:v6acl-1)# exit
ACOS(config)# ipv6 access-list v6acl-2
ACOS(config-access-list:v6acl-2)# permit ipv6 fc00:64::/96 any
ACOS(config-access-list:v6acl-2)# exit
```

The following commands configure the IPv4 NAT pools:

```
ACOS(config)# ip nat pool v4natpool-3 192.168.217.200 192.168.217.200
netmask /24
ACOS(config)# ip nat pool v4natpool-4 192.168.217.220 192.168.217.220
netmask /24
```

These commands access the configuration level for a virtual port on the VIP and configure the port to use the IPv4 pools:

```
ACOS(config)# slb virtual-server v6vip fc00:32::2020:2000
ACOS(config-slb vserver)# port 80 http
ACOS(config-slb vserver-vport)# access-list name v6acl-1 source-nat-pool
v4natpool-3
ACOS(config-slb vserver-vport)# access-list name v6acl-2 source-nat-pool
v4natpool-4
```

Each access-list command binds one of the IPv6 ACLs to the virtual port. The source-nat-pool option used with each command binds an IPv4 pool to the ACL. When the ACOS device receives a request for the VIP, the ACOS device matches the client address against the source addresses in the ACLs. ACOS then uses the IPv4 NAT pool bound to the first matching ACL.

ACOS translates the client's request from an IPv6 packet into an IPv4 packet. ACOS replaces the client's IPv6 address with an IPv4 address from the selected pool. The IPv6 VIP address is replaced with the server's IPv4 address.

If the client's address does not match the source address in any of the ACLs, the request is dropped.

NOTE: This is different from the behavior if a single NAT pool is used. If only one NAT pool is bound to the virtual port, the pool is used if the client's IP type (IPv4 or IPv6) is not the same as the IP type of the selected server. Otherwise, if the IP type of the client and the selected server is the same, SLB-PT is not required for the request. The request is sent to the server with the client's original IP address.

Multiple IPv4 and IPv6 Pools

It is not required to use pools of the same IP type as the IP type used by clients. For example, IPv6 pools are not required for IPv6 clients.

Using pools of the same IP type as the client IP type provides a way to control access to the real servers. When multiple pools are bound to a virtual port, the client's IP address must match the source address in at least one of the ACLs associated with the pools. Otherwise, the client's traffic is dropped.

NOTE: In the case of IPv4, IPv4 pools are still required if the VIP and the real servers are in different IPv4 subnets. For more information, see the "Source NAT for Servers in Other Subnets" section in the "Network Address Translation" section of the *System Configuration and Administration Guide*.

This example builds on the example in [Multiple IPv4 Pools](#). The virtual port will have 4 pools: 2 IPv4 pools and 2 IPv6 pools. Each of the IPv6 ACLs will be bound to an IPv4 pool and an IPv6 pool. If SLB selects an IPv4 server, the IPv4 pool bound to the ACL that matches the client's IP address will be used. Likewise, if SLB selects an IPv6 server, the IPv6 pool bound to the ACL will be used.

The following commands configure the IPv6 NAT pools:

```
ACOS(config)# ipv6 nat pool v6natpool-1 fc00:32::2020:2010
fc00:32::2020:2010 netmask 64
ACOS(config)# ipv6 nat pool v6natpool-2 fc00:32::2020:2020
fc00:32::2020:2020 netmask 64
```

The following commands bind the IPv6 NAT pools to the virtual port:

```
ACOS(config)# slb virtual-server v6vip fc00:32::2020:2000
ACOS(config-slb vserver)# port 80 http
```

```
ACOS(config-slb vserver-vport)# access-list name v6acl-1 source-nat-pool  
v6natpool-2  
ACOS(config-slb vserver-vport)# access-list name v6acl-2 source-nat-pool  
v6natpool-1
```


Proxy Protocol

Proxy Protocol is an internet protocol that conveniently transmits client information such as protocol, source IP, destination IP, source port, destination port, etc. by adding a small header into the TCP. The receiver can use such information to identify the actual client.

For more information on proxy protocol, refer [HAProxy](#).

This protocol supports two header formats:

- Version 1 - a human-readable (ASCII string) format
- Version 2 - a binary format

See the below examples:

- IPv4 proxy header ASCII string

```
PROXY TCP4 100.120.130.140.90.80.70.60 1234 80\r\n
```

- IPv6 proxy header ASCII string

```
PROXY TCP6 ffff:f...f:ffff...ffff:f...f:ffff 1234 80\r\n
```

ADC solution supports inserting proxy protocol header for both layer 4 and layer 7 with client address information. It helps you identify the client's IP address when the load balancer uses TCP or TCP proxy for back-end connections. The header insertion action is only with version 1 in its entirety and version 2 support up to the client headers information.

NOTE:

This feature is supported for TCP and TCP proxy virtual port and allows only IPv4 and IPv6 addresses. Also, proxy protocol can insert additional information after the address information, which is currently not supported.

CLI Configuration

Proxy protocol is configured with the 'proxy-header insert' command that is available through the following templates:

SLB TCP Configuration Template

These commands configure version 1 on the PROXY1 TCP template:

```
ACOS(config)# slb template tcp PROXY1
ACOS(config-l4 tcp)# proxy-header insert v1
ACOS(config-l4 tcp)# exit
```

These commands bind the PROXY1 TCP template to a virtual server.

```
ACOS(config)# slb virtual-server VIP1 10.1.1.1
ACOS(config-slb vserver)# port 80 tcp
ACOS(config-slb vserver-vport)# template tcp PROXY1
ACOS(config-slb vserver-vport)# exit
ACOS(config-slb vserver)# exit
ACOS(config)#
```

This feature has the following known limitations:

- Fast path and UDP is not supported.
- Following functionality or configuration is not supported:
 - aflex with TCP payload replace
 - Fast-fix with client IP insert
 - FTP virtual port

SLB TCP-Proxy Configuration Template

These commands configure version 1 on the PROXY1 TCP-proxy template:

```
ACOS(config)# slb template tcp-proxy PROXY1
ACOS(config-tcp proxy)# proxy-header insertv1
ACOS(config-tcp proxy)# exit
```

These commands bind the PROXY1 TCP-proxy template to a virtual server.

```
ACOS(config)# slb virtual-server VIP1 10.1.1.1
ACOS(config-slb vserver)# port 80 tcp-proxy
```

```
ACOS(config-slb vserver-vport)# template tcp-proxy PROXY1  
ACOS(config-slb vserver-vport)# exit  
ACOS(config-slb vserver)# exit  
ACOS(config)#
```

Part VI: Logging

This part contains topics pertaining to logging server load balancing activities.

The following topics are covered:

- [Web Logging for HTTP and RAM Caching](#)
- [Real-Time Logging for Failed Server Selection](#)

Web Logging for HTTP and RAM Caching

This section describes ACOS support of logging to external servers over TCP.

The following topics are covered:

Overview	954
Web Log String Customization	960

Overview

ACOS supports web logging for HTTP virtual ports, which can be used with HTTP load balancing or RAM caching. Web logging for load-balanced HTTP servers provides data about client access to servers. Web logging for RAM caching provides information about client access to content cached on the ACOS device. Web logging to external log servers is supported over TCP and UDP. Logging over TCP is applicable to web logging for HTTP virtual ports. The rest of this section describes this use of the feature.

Log Message Format

Web logs generated by the ACOS device use the following format:

```
Client-Src-IP -- Timestamp-on-ACOS-device Request-info Payload-length  
Referer-info User-agent Virtual-server-name:Virtual-port
```

Here is an example:

```
10.10.10.23 - - Mon Apr 23 23:38:13 2012 "GET / HTTP/1.0" 177  
"- " "Wget/1.12 (linux-gnu)" vs:80
```

This example uses a default log string. See [Web Log String Customization](#) for instructions on configuring custom log strings.

Configuring Web Logging

Configuring Log Webbing (Procedure)

To configure web logging:

1. Configure real servers for each log server. Add all ports receiving log messages to these configurations.
2. Add the log servers to a service group. Use round-robin load-balancing (the default method).
3. (Optional – TCP Ports only) Configure TCP-proxy templates to customize TCP settings for connections between the ACOS device and log servers.

For example, a TCP-Proxy template can be used to enable keepalive probes that ensure TCP connections with log servers remain established during idle periods between logs. (See CLI example.)

4. Configure a logging template. Add the service group containing the log servers to the template. Add the TCP-Proxy template to the service group as required.
5. To log web traffic sent to load-balanced HTTP servers, create an HTTP template and add the logging template to it.
6. To log web traffic served from the ACOS device's local RAM cache, create a RAM Caching template and add the logging template to it.
7. On the VIP, add the HTTP or RAM Caching template (or both) to the HTTP virtual port.

GUI Configuration

This section describes the GUI steps related to logging templates.

The configuration steps for the real servers, service groups, and VIPs are the same as without use of logging templates.

Configuring a Logging Template

1. Navigate to **ADC >> Template >> General** tab.
2. Click **Create** and select **Logging** from the drop-down menu.
3. Enter a name for the template.
4. Select the service group that contains the log servers.
5. If a custom TCP-proxy template for logging is configured, select it from the TCP Proxy list.
6. Click **OK**.

Applying a Logging Template to an HTTP Template

1. Navigate to **ADC >> Template >> L7 Protocols** tab.
2. Click **Create** and select **HTTP** from the drop-down menu.
3. Enter a name for the template.

4. Select the logging template from the drop-down list in the Logging Template field.
5. Click **OK**.

Applying a Logging Template to a RAM Caching Template

1. Navigate to **ADC >> Template >> Applications** tab.
2. Click **Create** and select **RAM Caching** from the drop-down menu.
3. Enter a name for the template.
4. Select the logging template from the drop-down list in the Logging Template field.
5. Click **OK**.

CLI Configuration

Configuring a Logging Template

To configure a logging template, use the `slb template logging` command to change the CLI to the configuration level for the template, where the `service-group` command is available to specify the name of the service group that contains the log servers.

NOTE: The minimum and maximum configurable values are platform specific. To check the configurable values, use the `show slb resource-usage` command.

This `template tcp-proxy` command specifies the name of the TCP-proxy template to use for managing TCP sessions between the ACOS device and the log servers.

Applying a Log Template to an HTTP Template

Use the `template logging` command at the configuration level for the HTTP template:

Applying a Log Template to a RAM Caching Template

Use the `template logging` command at the configuration level for the RAM Caching template:

Configuring Log Webbing (CLI Example)

The following commands configure web logging for an IPv4 virtual port and an IPv6 virtual port. On each virtual port, web logging is enabled both for HTTP load-balanced client-server traffic and for client access to content that is cached in the ACOS device's RAM cache.

In this example, two real servers are used as HTTP content servers and as logging servers. Clients send requests for HTTP content to port 80. ACOS either serves the requested from the local RAM cache, if available, or sends the request to one of the servers.

In this example, the ACOS device uses the same servers as the content servers and as the logging servers. Client requests for HTTP content are sent to port 80. Log traffic is sent to one of the following ports:

- 4999 – TCP port listening for log traffic sent over IPv4.
- 5999 – TCP port listening for log traffic sent over IPv6.

To begin, the following commands configure the servers:

```
ACOS(config)# slb server rs 10.10.10.11
ACOS(config-real server)# port 80 tcp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# port 4999 tcp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# port 5140 udp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
ACOS(config)# slb server rs6 3030::3011
ACOS(config-real server)# port 80 tcp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# port 5999 tcp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# port 6140 udp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
```

These commands configure service groups for the logging and application servers accessed by clients:

```
ACOS(config)# slb service-group logsg tcp
```

```
ACOS(config-slb svc group)# member rs 80
ACOS(config-slb svc group-member:80)# exit
ACOS(config-slb svc group)# exit
ACOS(config)# slb service-group udpsg udp
ACOS(config-slb svc group)# member rs 5140
ACOS(config-slb svc group-member:5140)# exit
ACOS(config-slb svc group)# exit
ACOS(config)# slb service-group v6logsg tcp
ACOS(config-slb svc group)# member rs6 80
ACOS(config-slb svc group-member:80)# exit
ACOS(config-slb svc group)# exit
ACOS(config)# slb service-group udpsg6 udp
ACOS(config-slb svc group)# member rs6 6140
ACOS(config-slb svc group-member:6140)# exit
ACOS(config-slb svc group)# exit
```

The following commands configure the TCP-proxy template, to enable keepalive probes:

```
ACOS(config)# slb template tcp-proxy logtcp
ACOS(config-TCP proxy template)# keepalive-probes 4
ACOS(config-TCP proxy template)# exit
```

The following commands configure the logging templates:

```
ACOS(config)# slb template logging logtemp
ACOS(config-logging)# service-group logsg
ACOS(config-logging)# template tcp-proxy logtcp
ACOS(config-logging)# exit
ACOS(config)# slb template logging udplog
ACOS(config-logging)# service-group udpsg
ACOS(config-logging)# exit
ACOS(config)# slb template logging logtemp6
ACOS(config-logging)# service-group v6logsg
ACOS(config-logging)# template tcp-proxy logtcp
ACOS(config-logging)# exit
ACOS(config)# slb template logging udplog6
ACOS(config-logging)# service-group udpsg6
ACOS(config-logging)# exit
```

The following commands configure the RAM caching templates:

```
ACOS(config)# slb template cache logcache
```

```
ACOS(config-ram caching)# min-content-size 20
ACOS(config-ram caching)# template logging udplog
ACOS(config-ram caching)# exit
ACOS(config)# slb template cache logcache6
ACOS(config-ram caching)# min-content-size 20
ACOS(config-ram caching)# template logging udplog6
ACOS(config-ram caching)# exit
```

The following commands configure the HTTP templates:

```
ACOS(config)# slb template http loghttp
ACOS(config-http)# template logging logtemp
ACOS(config-http)# exit
ACOS(config)# slb template http loghttp6
ACOS(config-http)# template logging logtemp6
ACOS(config-http)# exit
```

These commands configure the VIPs. The configuration of the snat and snat6 NAT pool referenced in the example is not included.

```
ACOS(config)# slb virtual-server vs 172.20.20.25
ACOS(config-slb vserver)# port 80 http
ACOS(config-slb vserver-vport)# source-nat pool snat
ACOS(config-slb vserver-vport)# service-group logsg
ACOS(config-slb vserver-vport)# template http loghttp
ACOS(config-slb vserver-vport)# template cache logcache
ACOS(config-slb vserver-vport)# exit
ACOS(config-slb vserver)# exit
ACOS(config)# slb virtual-server vs6 FC00::A2:103
ACOS(config-slb vserver)# port 80 http
ACOS(config-slb vserver-vport)# source-nat pool snat6
ACOS(config-slb vserver-vport)# service-group v6logsg
ACOS(config-slb vserver-vport)# template http loghttp6
ACOS(config-slb vserver-vport)# template cache logcache6
ACOS(config-slb vserver-vport)# exit
ACOS(config-slb vserver)# exit
```

Web Log String Customization

You can customize the structure and content of web log messages for requests sent to HTTP or HTTPS virtual ports. This capability customizes ACOS to efficiently log only the information that you require.

W3C Log Message Format

The CLI can modify web logging formats with the `format` command at the logging template configuration level. The logging template bound to the virtual server constructs log messages for HTTP/HTTPS requests according to the specified format.

For example, if the log message format is defined as shown below:

```
%{%Y-%m-%d %H:%M:%S}t %a %u %A %p %r %m %U %q %s "%{USER-AGENT}i" "%{REFERER}i" %H %v %b
```

Then the log message for an HTTP request is constructed as follows:

```
2018-02-08 19:30:53 11.0.0.40 2013-02-01 15:31:08 11.0.0.1 - 10.1.2.198 80
s1 GET /aa/cookie.htm - 200 "Wget/1.12 (linux-gnu)" "OK" HTTP/1.0 vip1
32494
```

In this log message, Feb 1 19:30:53 is the timestamp of when the log was received. The IP address of the server that received the log is 11.0.0.40. The remaining content of the log message is constructed according to the format string (defined by the `format` command that is configured within the logging template).

W3C Format Characters

[Table 23](#) describes W3C format characters supported on the ACOS device and references content in the example log message above.

Table 23 : W3C – Format Characters

Format Character	Description	Log Message Example
%%	Percent symbol	N/A
%a	Client IP address	11.0.0.1

Table 23 : W3C – Format Characters

Format Character	Description	Log Message Example
%A	Local IP address	10.1.2.198
%p	Port number that is used by the server for a request	80
%r	Name of the real server	s1
%m	HTTP request method	GET
%U	Requested URL path requested The log message does not include any query strings	/aa/cookie.htm
%q	Query string in a request	The second “-”
%u	For authenticated requests, the %u format character logs the remote user ACOS does not authenticate HTTP requests. Therefore, %u format character always return a null (-) value.	The first “-”
%s	HTTP status code for the request In RAM Caching deployments, the status code is not included in log messages; the %s format character returns a null (-) value in web logging messages.	200
%{time}t	Timestamp for when the request is processed For <i>time</i> you can specify the following: %Y – year %m – month	2013-02-01 15:31:08

Table 23 : W3C – Format Characters

Format Character	Description	Log Message Example
	%d – day %H – hour %M – minute %S – seconds	
%{USER-AGENT}i	User-agent sending the request	"Wget/1.12 (linux-gnu)"
%{REFERER}i	Referer of a request	"OK"
%H	HTTP request protocol	HTTP/1.0
%v	Name of the virtual server that processed the request Content for this value differs for virtual servers defined on shared or private partitions. For example, if the virtual server "vipv6" is defined in a partition named "l3v", then %v is parsed as "?l3v?vipv6" in the web log message.	Vip1
%b	Number of bytes in the response	32494

W3C Control Characters

In addition to the format characters described in [W3C – Format Characters](#), ACOS supports the following control characters:

- \\n – New line
- \\t – Carriage return (reset to the line beginning)
- \\r – Tab

W3C Format Consideration

If the format of a string includes an unsupported character, the log message will contain only the first section of valid information leading up to the unsupported character. Even if the log message contains supported content after the unsupported character, the latter section of supported content is not included in the log message. For example, given the structure below:

```
<supported "A"><unsupported "B"><supported "C">
```

The log message breaks at `<unsupported "B">`, displaying only content associated with `<supported "A">`.

For example, given the logging format `"%P%A%a%p"`, `"%P"` is not supported and as a result nothing is parsed into a log message.

For the logging format `"%p%P%a%A"`, the content after the unsupported `"%P"` format character is not included in the log message and the information for `"%p"` is the only content parsed into a log message.

To view which characters are parsed in a format string, use the `show slb template logging` command.

Do not use the question mark (?) as a literal character for log messages.

Configuring Web Logging Format

Perform the following commands to customize Web log messages.

GUI Configuration

1. Navigate to **ADC >> Template >> General** tab.
2. Click **Create** and select **Logging** from the drop-down menu.
3. In the Format field enter the series of up to 250 supported format characters. See [W3C – Format Characters](#) for information about format characters.
4. Click OK.

CLI Configuration

This example shows how to use the **format** command to create a log format, then use the **show slb template logging** command to verify the configuration:

```
ACOS(config)# slb template logging v-log
ACOS(config-logging)# format %a \n "%a"
ACOS(config-logging)# show slb template logging
slb template logging v-log
    format %a \n "%a"
ACOS(config-logging)#
```


Real-Time Logging for Failed Server Selection

The following topics are covered:

Overview	966
CLI Configuration	966
GUI Configuration	968

Overview

ACOS provides real-time notifications when the system detects a failed server selection event. Log entries includes event cause and users are notified of the instance. For example, real time logging can include connection limit or connection rate limit notification when the configured values are exceeded.

In addition to the system log entry, you can configure alerts using SNMP traps, enabling you to immediately respond to server selection failure.

To implement this feature, perform the following steps:

1. Within a server template, enable alerts for failed server selection and apply to the SLB server.
2. Apply the template to one or more real servers.
3. Add these servers to a service group, and apply the service group to a virtual server.
4. Configure SNMP traps for immediate event notifications of failed server selection. The SNMP trap notification will include information such as the server name, IP address, server port, partition name, and known cause for server selection failure.

CLI Configuration

Enter the following command to send SNMP traps if server selection fails:

```
ACOS(config)# snmp-server enable traps slb server-selection-failure
```

The following example shows the configuration of a server with enabled logging of failed server selection and SNMP notification alerts.

```
ACOS(config)# slb template server svtempl  
ACOS(config-rserver)# conn-rate-limit 1  
ACOS(config-rserver)# log-selection-failure  
ACOS(config-rserver)# exit
```

Create a server and apply the template with logging enabled for failed server selection.

```
ACOS(config)# slb server srv266 10.10.100.10
ACOS(config-real server)# template server svtempl
ACOS(config-real server)# port 80 tcp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
```

Create a service group and add the srv266 server.

```
ACOS(config)# slb service-group fail-servers tcp
ACOS(config-slb svc group)# member srv266 80
ACOS(config-slb svc group-member:80)# exit
ACOS(config-slb svc group)# exit
```

Create a virtual server:

```
ACOS(config)# slb virtual-server fail-233 1.1.1.1
ACOS(config-slb vserver)# port 80 http
ACOS(config-slb vserver-vport)# name _1.1.1.1_HTTP_80
ACOS(config-slb vserver-vport)# service-group fail-servers
ACOS(config-slb vserver-vport)# exit
ACOS(config-slb vserver)# exit
```

Enable SNMP notifications for failed server selection:

```
ACOS(config)# snmp-server enable traps slb server-selection-failure
ACOS(config)# snmp-server host 192.168.3.67 version v2c public udp-port
162
```

The following is an example of show command output after an instance of failed server selection:

```
ACOS# show log
Log Buffer: 30000
Dec 28 2012 07:09:15 Warning [ACOS]:Server srv266(port:80) selection
failure(exceeded_real_server_connection_rate_limit)
Dec 28 2012 07:09:15 Warning [ACOS]:Server srv266(port:80) selection
failure(exceeded_real_server_connection_rate_limit)
```

GUI Configuration

Perform the following procedures to configure real-time alerts with the GUI:

Configuring Logging

1. Navigate to **ADC >> Templates >> SLB** tab.
2. Click **Create** and select **Server** from the drop-down menu.
3. Enter a name and other information for the template. (See online help for details on mandatory fields.)
4. Select checkbox in the **Log Selection Failure** field.

This option enables real-time logging for the server selection failure event.

5. Modify optional template configuration as required. Available real time logging events include *Connection Limit* and *Connection Rate Limit*.
6. Apply the server template to one or more real servers.

This provides warning messages that are logged when a threshold is exceeded.
7. Click **OK**, or click **Update** if you are modifying an existing template.

Configuring SNMP Traps

1. Navigate to **System >> Getting Started**.
2. In the SNMP section, click the **Edit** (Pencil) button.
3. For System SNMP Service, select the **Enable** radio button.
4. In the Trap List section, select the **Server Selection Failure trap** in the **SLB Group** section.
5. Update the other fields.
6. Click **Configure SNMP**.

Glossary

C

Compound Health Monitor

A health-check construct in ACOS that chains multiple probes (such as TCP and HTTP content checks) with Boolean logic to validate service health before traffic steering.

Connection Reuse

An ACOS feature that multiplexes multiple client requests over existing server-side TCP connections to reduce handshake overhead and improve server efficiency.

D

DNS Cache Sharing

An ACOS feature that shares DNS cache entries across processes or contexts to reduce upstream lookups and accelerate DNS responses at scale.

DNS Firewall Support

An ACOS DNS security capability that enforces filtering, such as blocklists and response manipulation, against malicious or unwanted domains at the resolver or forwarder layer.

DoH

An ACOS implementation of DNS over HTTP (DoH) that uses templates to proxy or terminate encrypted DNS transactions and apply DNS policies.

DOHVIP

A VIP entity in A10 Networks that supports DNS-over-HTTP(S) services by binding DoH templates, service groups, and policies for encrypted DNS traffic.

Dynamic Real Servers

An ACOS mechanism that programs real-server

entries dynamically, such as through DNS-driven discovery, to scale backend services without static configuration.

E

Explicit HTTP Proxy

An ACOS forward-proxy mode that explicitly terminates client HTTP and CONNECT traffic and applies policies such as URL filtering, category controls, and SSL/TLS handling.

G

GLID

A Global Limit ID (GLID) that provides a shared rate-limit bucket for multiple A10 policies to coordinate consistent throttling at scale.

GLM

A Global Licensing Manager (GLM) service from A10 Networks that

activates and manages ACOS feature licenses and throughput tiers across deployments.

L

LID

A Local Limit ID (LID) that provides a per-context rate-limit identifier distinct from GLID-based limits in ACOS.

N

Next Hop Load Distributor

A feature in A10 Networks that load balances routed egress traffic across multiple next-hop gateways to improve path utilization and availability.

R

RAM Cache

An in-memory ACOS web-object cache that accelerates HTTP responses, reduces

origin-server load, and can be tuned for specific content types and cacheability rules.

S

Service Group

A container in ACOS that groups real servers and ports and defines health-monitoring methods, load-balancing logic, and member-selection policies.

SPE

A Security and Policy Engine (SPE) variant in A10 Networks' hardware or software that accelerates crypto, L4–L7 processing, and security functions on Thunder platforms.

Stateless Auto Switch

An ACOS automation feature that switches to stateless selection when conditions or policies require it to maintain

availability and scale without state tracking.

Stateless Load Balancing

A stateless load-balancing mode in ACOS that forwards traffic without per-flow state to increase scale and throughput while shifting some responsibilities to endpoints.

Strict Transaction Switching

A load-balancer behavior in A10 Networks that keeps a client request bound to the same server for the duration of a transaction, ensuring correctness for applications requiring affinity.

T

TCS

A Transparent Cache Switching (TCS) feature in A10 Networks that steers HTTP traffic to cache proxies based on policy to enable offload and performance gains

without client-side configuration.

Transparent Proxy Configuration

A transparent (implicit) ACOS proxy mode that intercepts client traffic in-path and applies policies or acceleration without requiring client proxy settings.

U

URL Hash First

A URL-hashing variant in ACOS that uses the beginning portion of the URL for hash computation to improve persistence for path-structured content.

URL Hash Last

A URL-hashing variant in ACOS that uses the trailing portion of the URL for hash computation, useful when object identifiers appear at the end of paths.

URL Hash Offset

An ACOS control that adjusts the substring offset used for URL-hash computation to tune persistence behavior for certain URL patterns.

URL Hash Persist

A persistence method in A10 systems that uses URL hashing to determine server stickiness, improving cache locality and application affinity based on object paths.

URL Redirect Rewrite

An ACOS capability that modifies HTTP redirects in real time to ensure correct routing and application behavior through proxies or load balancers.

V

VIP Source NAT

A source-NAT capability in A10 Networks that uses the VIP address or

pool to simplify return-path routing and ensure symmetric flow handling in SLB deployments.

features with minimal redesign.

Virtual port

A port-level construct under an ACOS Virtual Server that maps a protocol and port to policies, templates, health checks, and load-balancing logic.

Virtual server

A configuration object in ACOS that represents the client-facing VIP and application services, binding service groups, templates, policies, and features to deliver the application.

Virtual Wire Mode

A transparent L2/L3 mode in A10 systems that allows ACOS to operate inline without changing IP addressing, enabling insertion of security and SSL/TLS



Contact Us

©2026 A10 Networks, Inc. All rights reserved. A10 Networks, the A10 Networks logo, ACOS, A10 Thunder, Thunder TPS, A10 Harmony, SSLi and SSL Insight are trademarks or registered trademarks of A10 Networks, Inc. in the United States and other countries. All other trademarks are property of their respective owners. A10 Networks assumes no responsibility for any inaccuracies in this document. A10 Networks reserves the right to change, modify, transfer, or otherwise revise this publication without notice. For the full list of trademarks, visit: www.a10networks.com/company/legal/trademarks/.