ACOS 6.0.8
Configuring Overlay Networks

December, 2025

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

Introduction

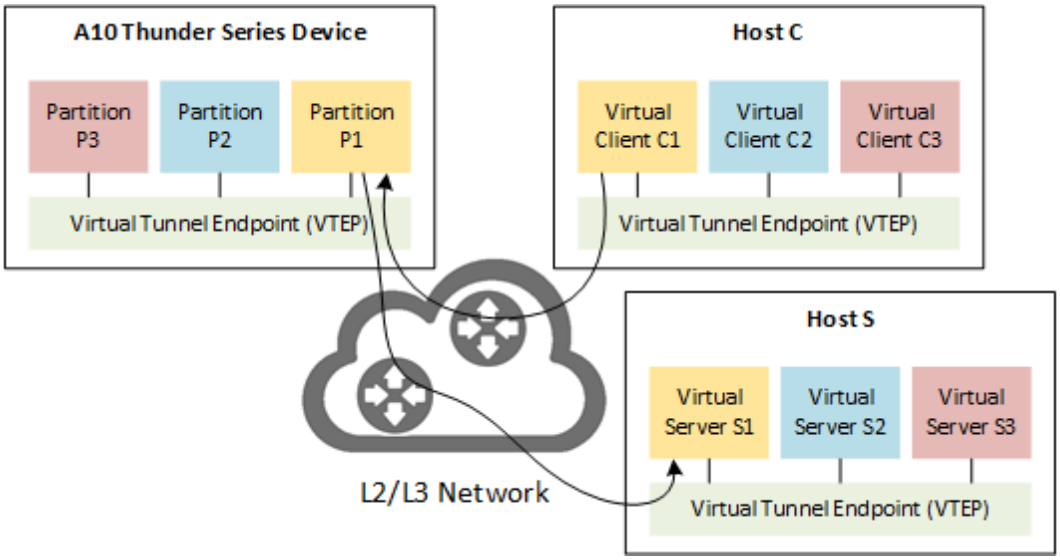
An overlay network is a virtual network built on top of an existing physical network in order to introduce a service that is not available in that existing network. The overlay network uses virtual links to connect various nodes of the existing physical network. Virtual network traffic is then tunneled across the network using technologies like VXLAN (Virtual Extensible LAN) and NVGRE (Network Virtualization using Generic Routing Encapsulation).

Both VXLAN and NVGRE serve to connect multiple Layer 3 networks and give the appearance that they all share the same Layer 2 domain.

Example

Consider the illustration in Figure 1:

Figure 1 : Overlay Network



- Host C, Host S and a Thunder Appliance are connected to each other through an overlay or tenant network. Packets are sent through the existing underlay or provider network (labeled “L2/L3 Network”), but the tunnel endpoints in the

overlay are responsible for properly encapsulating and decapsulating packets; the configuration of the underlay network remains unchanged.

- The tunnel endpoints mark the edge of the overlay network, and the virtual servers and clients represent portions of the network being extended on top of the overlay network.
- The virtual appliances behind each tunnel endpoint are grouped together by segment. Each segment is uniquely identified by a 24-bit segment identifier; VXLAN calls this a VXLAN Network Identifier (VNI), and NVGRE refers to this as a Virtual Subnet Identifier (VSID). All virtual machines on a segment are uniquely identified by their VNI/VSID and MAC address, and only virtual machines on the same VNI/VSID are able to communicate with each other.
- In the example above, the servers and clients in segment 100 can communicate with each other through the overlay tunnel; however, they are not able to communicate with the servers and clients in the other segments.
- Some addressing benefits of such a topology are summarized below:
- Isolated end-to-end network domains for tenants
- Because the segments are uniquely identified and isolated from each other, each network domain (segment) is able to be configured for specific needs (for example, network traffic, network security, bandwidth, or other specialized policies).
- IP addresses, MAC addresses, and VLANs can be reused across tenants
- Also because of the network (segment) isolation described above, it is possible to have duplicate IP addresses, MAC addresses, and VLANs across different segments, but not within the same segment.
- Tenant addressing is de-coupled from physical network addressing
- The physical underlay network addressing and configuration are not modified. Encapsulated packets travel through the overlay network via the tunnel endpoints, and only pass through the existing physical underlay network. Thus, the only network addressing configuration that needs to be performed is in the overlay network.
- Lower overhead within the existing physical network.
- Scaling benefits - a high number of virtual networks can be configured, compared to VLANs

- Both VXLAN and NVGRE insert a new 24-bit identifier (VNI for VXLAN, or VSID for NVGRE) into each IP packet, meaning that over 16 million L2 logical networks can be created, compared to the 4,094 limit with traditional VLANs.
- Provides MAC addresses table scaling in the physical network

Configuring a Software Defined Network

This section describes how to configure and monitor overlay tunnels to create a Software Defined Network (SDN) or Overlay Network.

The following topics are covered:

- Understanding Packet Encapsulation and Transport
- Configuring a Software Defined Network
- Understanding Adhoc Tunnels
- Configuring a Sample Adhoc Tunnel

Understanding Packet Encapsulation and Transport

The following topics are covered:

Understanding VXLAN Encapsulation

In a VXLAN network, Layer 2 packets are tunneled over a Layer 3 network using UDP/IP over IP frames, as shown in Figure 2.

Figure 2 : VXLAN Packet Encapsulation

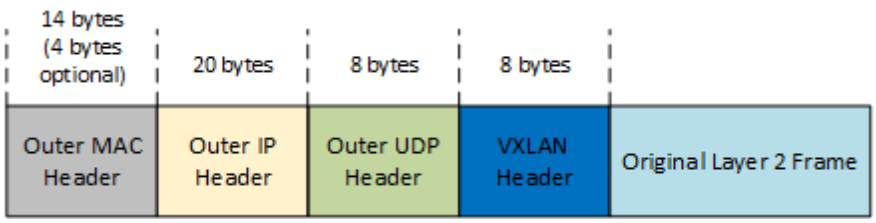


Table 1 describes the various fields in the VXLAN encapsulation.

Table 1 : VXLAN Packet Encapsulation

Field	Description
Outer MAC Header	Contains the following: • Destination MAC address (48 bits) • Source MAC address (48 bits) • VLAN type set to 0x8100 (16 bits) • VLAN ID (16 bits) • Ethertype of 0x0800 to designate the payload as an IPv4 packet (16 bits)
Outer IP Header	Contains the following • IP header miscellaneous data (72 bits) • Protocol set to 0x11 to indicate UDP (8 bits) • Header chksum (16 bits) • Source IP address of the originating VTEP (32 bits) • Destination IP address of the target VTEP (32 bits)
Outer UDP Header	Contains the following: • Source port (16 bits) • VXLAN port (16 bits) • UDP length (16 bits) • UDP checksum (16 bits)
VXLAN Header	Contains the following • VXLAN flags Reserved bits set to zero except for the third bit, which is set to 1 to indicate a valid VNI. (8 bits) • VNI (24 bits) Two sets of fields, 24 bits and 8 bits, that are reserved and set to zero.

Understanding NVGRE Encapsulation

NVGRE packets are tunneled directly over IP (as opposed to UDP in VXLAN). The outer IP header protocol field is set to 47 (the protocol number for GRE).

Figure 3 : NVGRE Packet Encapsulation

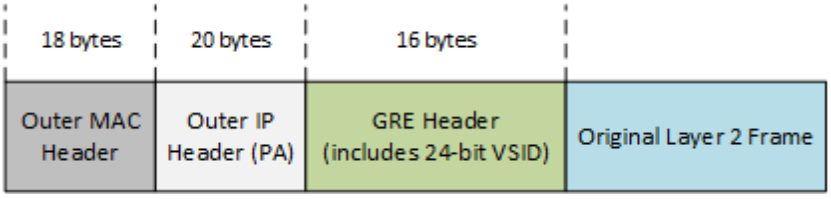


Table 2 describes the various fields in the NVGRE encapsulation.

Table 2 : NVGRE Packet Encapsulation

Field	Description
Outer MAC Header	The source and destination MAC addresses.
Out IP Header	The provider addresses (PA); the source and destination IP addresses for the tunnel.
GRE Header	Contains the following: • Flag field (8 bits) The third bit (the “key” flag) is set to 1, meaning the header contains a valid VSID. The remaining bits are set to zero. • Protocol Type This field is set to 0x6558 which is reserved for Transparent Ethernet Bridging. • 24-bit VSID: These 24 bits in the 32-bit key field in the GRE header. designate an individual NVGRE network on which virtual machines can communicate. Virtual machines in different NVGRE networks cannot communicate with each other.

Table 2 : NVGRE Packet Encapsulation

Field	Description
	• Flow ID: These last 8 bits of key field are optional and can be used to add per-flow entropy within the same VSID so that entire key field (32 bits) can be used for Equal-Cost Multi-Path (ECMP) routing purposes by switches and routers. If Flow ID is not generated, it must be set to zero.

Understanding GRE Encapsulation

ACOS GRE tunneling supports IPv4 GRE tunneling with IPv4 and IPv6 payload. The GRE KeepAlive feature is also introduced to continuously monitor GRE tunnel endpoints' health and bring down the corresponding logical interface (LIF) upon failure. The Encapsulation and Decapsulation support of the GRE packets is configured with the `encap GRE` CLI command.

The following features are provided:

- Send Syslog and SNMP trap alerts when the GRE tunnel is up or down
- View the GRE tunnel status using CLI and aXAPI
- Distribute routes based on GRE tunnel status

Known Limitations

The following functionalities are not supported in this release:

- Fragmentation for ISIS packets
- IPv6 GRE tunnels
- GRE supported for non-FTA models only

Understanding Fragmentation of Tunneled Packets

As a result of the additional header (50 bytes for VXLAN and 42 Bytes for NVGRE) added to tunneled packets, fragmentation of the packet can occur if the host is not aware of this condition and the maximum segment size (MSS) is exceeded. The ACOS

device will automatically adjust the MSS exchanged for TCP flows to take into account this header.

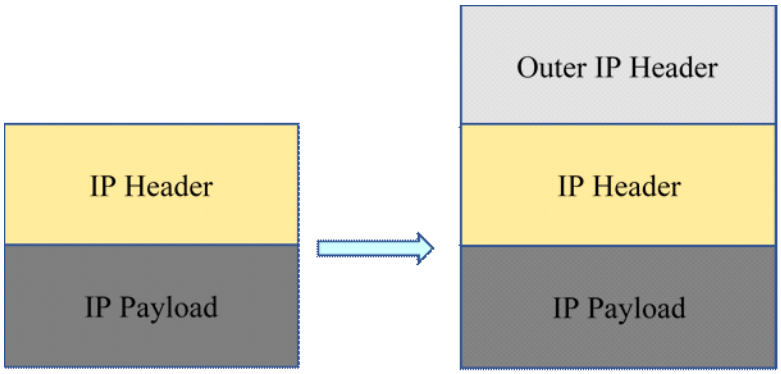
To disable this automatic adjustment, use the **`overlay-tunnel options tcp-mss-adjust-disable`** command. For more information, refer to the **`overlay-tunnel`** command in the Command Line Interface Reference.

Packet fragmentation for tunneled packets is not supported on FTA platforms. If you are configuring overlay tunnels on an FTA device, make sure to enable this automatic adjustment to avoid fragmented packets being dropped. Use the following command: `no overlay-tunnel options tcp-mss-adjust-disable` .This command only works for TCP traffic. For non-TCP traffic, you must manually adjust the MTU on your device to avoid fragmentation.

Understanding IP Encap (IPinIP) Tunneling

The IP encap supports IPinIP, IPinIPv6, IPv6inIP and IPv6inIPv6 encapsulations. Payloads of IPv4 and IPv6 can be transported atop IPv4 or IPv6 encapsulation headers. Encapsulation and Decapsulation support of the IPv4 and IPv6 encapsulation is configured with the `ip-encap` VTEP encap.

Figure 4 : IP Encap (IPinIP) Tunneling

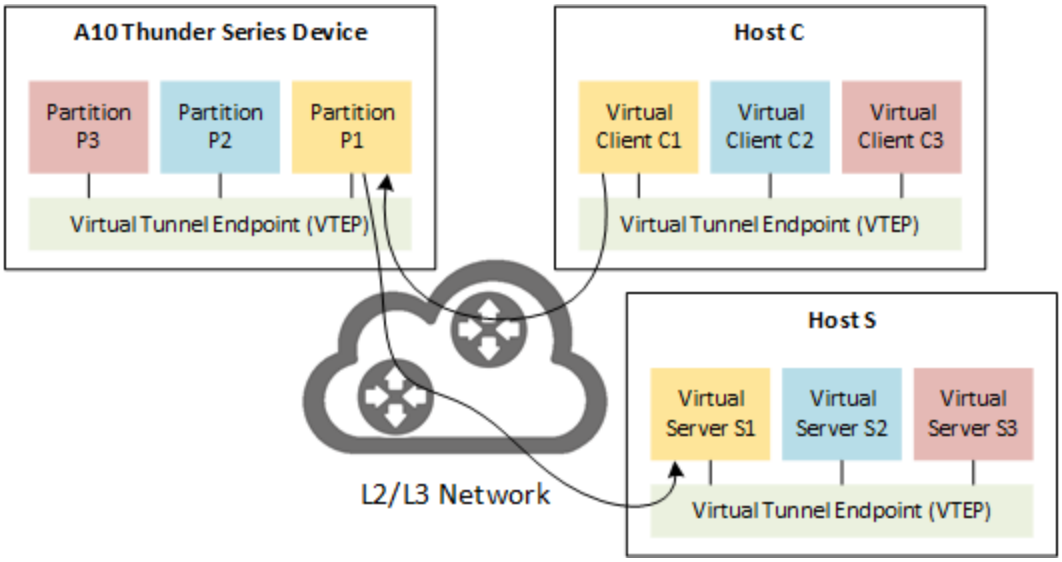


Configuring a Software Defined Network

This section describes how to configure the overlay tunnels to create a Software Defined Network (SDN) or Overlay Network.

Figure 5 shows a sample overlay tunnel topology that will be used as a sample configuration in this section.

Figure 5 : Sample Overlay Tunnel Topology



NOTE: The instructions assume that you have already created the necessary L3V partitions on your system. The overlay partition should be an L3V partition.

The following topics are covered:

Configuring the Underlay/Provider Partition

This section provides the commands used to configure the provider partition. More details about each command are available in the Command Line Interface Reference.

This example is configured using VXLAN encapsulation; replace **`encap vxlan`** with **`encap nvgre`** if you want to use NVGRE encapsulation.

The following topics are covered:

Beginning the Tunnel Configuration and Specifying the Encapsulation

Enter configuration mode on your ACOS device and enter the following commands.

```
ACOS# config
ACOS(config)# overlay-tunnel vtep 1
ACOS(config-overlay-tunnel:1)# encap vxlan
```

Configuring the Source IP Address

These commands create the binding between the user partitions (P1 and P2) and the tunnel endpoint VTEPa in the provider partition.

NOTE: The source or local IP address used in the configuration (in this case, 120.1.1.10) needs to be present in the underlay partition as an interface IP or a floating IP before any traffic will pass through.

```
ACOS(config-overlay-tunnel:1)# local-ip-address 120.1.1.10
ACOS(config-overlay-tunnel:1-src-vtep)# vni 100 partition p1 lif 10
ACOS(config-overlay-tunnel:1-src-vtep)# vni 200 partition p2 lif 20
ACOS(config-overlay-tunnel:1-src-vtep)# exit
ACOS(config-overlay-tunnel:1)#
```

Creating the Remote VTEP Configuration for the Clients (VTEPc)

These commands create the mapping for packets between VTEPa to VTEPc, identifying the destination or remote IP address for VTEPc and the fact that there are two VNIs behind VTEPc:

```
ACOS(config-overlay-tunnel:1)# remote-ip-address 100.1.1.10
ACOS(config-overlay-tunnel:1-dst-vtep)# vni 100
ACOS(config-overlay-tunnel:1-dst-vtep)# vni 200
ACOS(config-overlay-tunnel:1-dst-vtep)# exit
ACOS(config-overlay-tunnel:1)#
```

Creating the Remote VTEP Configuration for the Servers (VTEPs)

These commands create the mapping for packets between VTEPa to VTEPs, identifying the destination or remote IP address for VTEPs and the fact that there are two VNIs behind VTEPs:

```
ACOS(config-overlay-tunnel:1)# remote-ip-address 110.1.1.10
ACOS(config-overlay-tunnel:1-dst-vtep)# vni 100
ACOS(config-overlay-tunnel:1-dst-vtep)# vni 200
ACOS(config-overlay-tunnel:1-dst-vtep)# exit
ACOS(config-overlay-tunnel:1)#
```

Manually Configuring the Host-to-Destination VTEP Mapping to Avoid Dynamic Learning

By default, the ACOS device will attempt to dynamically learn the host-to-destination VTEP mapping by using unicast IP and ARP response packets coming over the tunnel.

The `host` command is used to manually map the source VTEP to the logical interface behind the destination VTEP. For example, the following command maps VTEPa to Client C1 (IP address 10.1.1.10 and MAC address 001c.011c.111c) on VNI 100 behind VTEPc (IP address 100.1.1.10):

```
ACOS(config-overlay-tunnel:1)# host 10.1.1.10 001c.011c.111c vni 100
remote-vtep 100.1.1.10
```

Additional mappings between VTEPa to Client C2, Server S1, and Server S2 may be similarly configured:

```
ACOS(config-overlay-tunnel:1)# host 20.1.1.10 002c.022c.222c vni 200
remote-vtep 100.1.1.10
ACOS(config-overlay-tunnel:1)# host 10.1.1.20 001s.011s.111s vni 100
remote-vtep 110.1.1.10
ACOS(config-overlay-tunnel:1)# host 20.1.1.20 002s.022s.222s vni 200
remote-vtep 110.1.1.10
```

Configuring the Overlay/Tenant Partition P1

The following topics are covered:

Configuring a Standalone LIF as a Layer 3 Interface

The following commands configure a standalone Layer 3 logical tunnel interface to be used for outbound traffic from partition P1 to VTEPa:

```
ACOS[p1](config)# interface lif 10
ACOS[p1](config-if:lif10)# ip address 10.1.1.1 /24
ACOS[p1](config-if:lif10)# exit
ACOS[p1](config)#
```

Configuring a Pure Layer 2 LIF

Similarly, you can configure an untagged Layer 2 logical interface in a VLAN; an example of this is shown with partition P2:

```
ACOS[p2](config)# vlan 20
ACOS[p2](config-vlan:20)# untagged lif 20
ACOS[p2](config-vlan:20)# tagged ethernet 5
ACOS[p2](config-vlan:20)# exit
ACOS[p2](config)#
```

In such a configuration, when a packet is received on VLAN 20, a copy is tunneled over to the VTEP, and a second copy is sent to ethernet interface 5.

Configuring a LIF as a Layer 2 Interface Inside a VLAN with a Layer 3 Virtual Ethernet Endpoint

Alternatively, you can make the Layer 2 LIF into a Layer 3 interface by adding the IP address to a virtual ethernet interface configured on the VLAN:

```
ACOS[p1](config)# vlan 20
ACOS[p1](config-vlan:20)# untagged lif 20
ACOS[p1](config-vlan:20)# tagged ethernet 5
ACOS[p1](config-vlan:20)# router-interface ve 20
ACOS[p1](config-vlan:20)# exit
ACOS[p1](config)# interface ve 20
ACOS[p1](config-if:ve20)# ip address 20.1.1.1 /24
ACOS[p1](config-if:ve20)# exit
```

```
ACOS[p1](config)#
```

Verifying the Configuration

To verify your configuration, you can use the following **show** commands. Additional information for each command is available in the *Command Line Interface Reference*.

To verify the configuration of the VTEP, use the show running-config overlay-tunnel command. For more information, see , see *Command Line Interface Reference.*

```
ACOS# show running-config overlay-tunnel
!
overlay-tunnel vtep 1
  local-ip-address 120.1.1.10
   vni 100 partition p1 lif 10
  remote-ip-address 100.1.1.10
   vni 100
!
```

Additional Implementation Notes

The following notes clarify the current implementation of SDN.

- This feature is supported on all A10 Thunder Series platforms and hypervisors (KVM, ESXi, Hyper-v and OCI, and OpenStack) except for the following:
  - A10 Thunder Series 5630
  - vThunder platforms
    - AWS
    - Azure
- Only IPv4 networks, hosts and tunnels are supported, both for the overlay and the underlay. All non-IPv4 traffic going across the overlay will need the corresponding logical interface to be configured as a Layer 2 interface.
- Any Layer 3 or Layer 4 features configured for the overlay traffic needs to be explicitly configured on the logical interface. For example if access-list is required to police incoming traffic, it will need to be bound to the logical interface.

Configuring it and binding it on the underlay interface will not apply to the overlay traffic as the incoming traffic is de-encapsulated first and then the policy lookup is performed.

- For VRRP-A, the VTEP IP should be configured as a floating IP. With VRRP-A configured, all VTEP and logical interface partitions need to be part of the same VRRP group. The VTEP IP should be configured as the vrid-lead, while the vrid resources in the logical interface partition, will need to be configured to follow the VTEP IP.
- With VRRP-A configured, the provider partition that has the VTEP configured and all related logical interface (customer) partitions must be configured so that all of them are active or standby on the same ACOS device. VTEP-to-logical interface communication between different ACOS devices is not supported.
- Forward reference for the logical interface partition as well as the logical interface is supported for the overlay-tunnel configuration. For example, the overlay-tunnel configuration can be added without the need to first configure the overlay partition (if different from the underlay partition) or creating the logical interface.
- Bridge VLANs with logical interfaces configured as Layer 2 interfaces are not supported.
- If the real servers are across an overlay tunnel, then the health check packets from both Active and Standby ACOS Thunder devices will go over the overlay tunnel. Responses from the servers will only be received by the Active ACOS device with the active Floating VTEP IP, so the real servers on the Standby will be in “down” state until switchover happens.
- Given that both Active and Standby ACOS devices will be sending health checks to the servers, if these were to go over the tunnel, the responses are directed only to the Active ACOS device on the VTEP IP address. On the Standby ACOS device, the Servers will be in “down” state until the switchover happens and health checks start passing.

Understanding Adhoc Tunnels

ACOS supports both point-to-point and point-to-multipoint tunnels. In a point-to-point tunnel, a Virtual Private Network (VPN) is set up between a Thunder device and a single destination. In a point-to-multipoint tunnel, there are multiple destinations connected to a Thunder device.

In a point-to-multipoint tunnel configuration, there are large number of remote tunnels and their remote IP addresses are unknown to ACOS. It is practically not possible to configure all the remote tunnel endpoints. Therefore, ACOS dynamically creates a single tunnel using GRE and IPinIP encapsulation without specifying a remote (or source) IP address for every tunnel endpoint. Such a tunnel is called an 'adhoc' tunnel.

For an adhoc tunnel configuration:

- A dedicated unique Logical Interface (LIF) is configured to hold multiple remote tunnel endpoints.
- A local IP address is configured for every remote tunnel endpoint and every overlay tunnel in the partition.
- A class-list can be bound to the remote IP address.

The adhoc tunnel always originates from the remote tunnel endpoint and ends at the local IP address. Each endpoint connected to the Thunder device through an adhoc tunnel has a unique IP address.

Configuring a Sample Adhoc Tunnel

This section provides the commands used to configure the adhoc tunnel. For more details about each command, see the *Command Line Interface Reference*.

The following topics are covered:

Configuration Overview

An adhoc tunnel has the following:

- It can be configured with or without a class-list and must be pre-configured.
- It has a LIF interface and it must be pre-configured.
- It is defined in an overlay tunnel that contains an encapsulation type (gre and ip-encap), a local IPv4 or IPv6 address, and a default (zero) remote IPv4 or IPv6 address with or without the classlist and a LIF interface. The remote IP address must not be a non-zero IP address.

The following example is configured with a class-list and GRE encapsulation type:

Configuring a Class-list

The following commands configure a class-list to bind the internal subnet with LSN LID.

```
ACOS(config)# class-list INTERNAL
ACOS(config-class list)# 99.0.1.0/24 lsn-lid 1
ACOS(config-class list)# 172.16.231.0/24 lsn-lid 1
ACOS(config-class list)# exit
```

Configuring a LIF interface

The following commands configure a LIF interface.

```
ACOS(config)# interface lif 100
ACOS(config-if:lif:1)# exit
```

Configuring an Overlay tunnel

The overlay tunnel is created with GRE encapsulation, a local IP address, and a default (zero) remote IP address using the pre-defined class-list filter and LIF interface.

```
ACOS(config)# overlay-tunnel vtep 1
ACOS(config-overlay-tunnel:1)# encap gre
ACOS(config-overlay-tunnel:1)# local-ip-address 10.10.10.135
ACOS(config-overlay-tunnel:1-src-vtep:10.10.10.135)# exit
ACOS(config-overlay-tunnel:1)# remote-ip-address 0.0.0.0 class-list
INTERNAL
ACOS(config-overlay-tunnel:1-dst-vtep:0.0.0.0)# lif 100
ACOS(config-overlay-tunnel:1-dst-vtep:0.0.0.0)# exit
ACOS(config-overlay-tunnel:1)# exit
```

Verifying the Configuration

The following command shows the running configuration:

```
ACOS(config)# show running-config
class-list INTERNAL
  99.0.1.0/24 lsn-lid 1
  172.16.231.0/24 lsn-lid 1
!
interface lif 100
!
overlay-tunnel vtep 1
  encap gre
  local-ip-address 10.10.10.135
  remote-ip-address 0.0.0.0 classlist-filter INTERNAL
    lif 100
!
```

The following command shows the overlay tunnel encapsulation table information for the adhoc tunnel:

```
ACOS(config)# show overlay-tunnel encap-table
IP encap endpoints
------------------
Local endpoint  Remote endpoint  Encap type   Static  Age    Vtep ID   LIF

--------------------------------------------------------------------------------
--------
10.10.10.135        10.1.1.100       GRE          NO     8      100
100     10.10.10.135        10.1.1.101       GRE          NO     0
100       200
```

The following command shows the ARP table entries for the adhoc tunnel:

```
ACOS(config)# show arp 61.61.61.100
Total arp entries: 1            Age time: 300 secs
IP Address      MAC Address       Type      Age    Interface    Vlan/Encap
-------------------------------------------------------------------------
61.61.61.100    021f.a000.00f8    Dynamic   40     lif 100          1

vtep info
---------
vtep id: 100  age: 45  partition id: 0     static: No    encap type: GRE
local: 30.1.1.2         remote: 10.1.1.100
```

The following command shows the IPv6 neighbor entries for the adhoc tunnel:

```
ACOS(config)# show ipv6 neighbor
Total IPv6 neighbor entries: 4
IPv6 Address              MAC Address    Type    Age  State      Interface
 Vlan/Encap
-------------------------------------------------------------------------------
-------------
6060::100                 021F.A000.00F8 Dynamic 201  Reachable  lif 100
        1
fe80::21f:a0ff:fe0f:bb82 001F.A00F.BB82 Dynamic 209  Stale      Management
        1
3010::1                   001F.A040.873C Dynamic 83   Reachable  ve10
        10
2020::100                 9EE2.C491.C75A Dynamic 82   Reachable  ve20
        20
```

Contact Us