



Configuring PPPoE over L2TPv3 Tunnels

The Layer 2 Tunneling Protocol Version 3 (L2TPv3) defines the L2TP protocol for tunneling Layer 2 payloads over an IP core network. In this feature, PPPoE packets are transported between network sites. The IP packets carry L2TP data messages and PPPoE sessions terminate on a Broadband Network Gateway (BNG) instead of at each Point of Presence (PoP). This feature allows you to extract and terminate PPP sessions for incoming traffic encapsulated in L2TPv3 IPv6 tunnels.

- [Information About Configuring PPPoE over L2TPv3 Tunnels, on page 1](#)
- [How to Configure PPPoE over L2TPv3 Tunnels, on page 7](#)
- [Additional References for Configuring PPPoE over L2TPv3 Tunnels, on page 27](#)
- [Feature Information for Configuring PPPoE over L2TPv3 Tunnels, on page 28](#)

Information About Configuring PPPoE over L2TPv3 Tunnels

Overview of PPPoE over L2TPv3 Tunnels

The PPPoE over L2TPv3 feature allows you to establish PPPoE sessions for incoming traffic using Layer 2 Tunneling Protocol Version 3 (L2TPv3) IPv6 tunnels. An L2TPv3 over IPv6 tunnel is a static/stateless P2P overlay tunnel between a physical edge/aggregation router and its peer entity. The peer entity is typically a virtual Broadband Network Gateway (vBNG), Virtual Network Function (VNF). The L2TPv3 tunnel transports ethernet traffic to and from CPEs. Each CPE is connected to an Access Node Optical Line Terminal (OLT)/Digital Subscriber Line Access Multiplexer (DSLAM).

Point-to-Point Protocol over Ethernet (PPPoE)—protocol describing the encapsulation of PPP frames inside ethernet frames and tunneling packets over Digital Subscriber Lines (DSLs) to Internet Service Providers (ISPs).

L2TPv3 Tunnel Interface—logical interface for terminating Broadband Subscriber Layer 2 Ethernet attachment circuits (port/VLAN) from access or edge routers over an IPv6 network with L2TPv3 encapsulation for BNG services. For further information, see IETF [RFC8159](#).

The difference between the situation before and after the introduction of PPPoE sessions for L2TPv3 tunnels is shown in Figures 1 and 2 in: [Overview of PPPoE over L2TPv3 Tunnels—Example Topology, on page 2](#).

You can use Point-to-Point Protocol over Ethernet (PPPoE) sessions via EoL2TPv3oIPv6 tunnels to deliver the same functions as those described in: [How to Enable and Configure PPPoE on Ethernet](#).

The PPPoE sessions used by this feature have the following key characteristics:

- H-QoS shaper per-session
- In/Out Access Control Lists (ACLs)
- Dual Stack
- Ingress QoS Policing
- Unicast Reverse Path Forwarding (uRPF)
- Lawful Intercept (LI)—both Radius & SNMP-based
- Session termination in VRF

The scope of a vBNG on a static/stateless P2P EoL2TPv3oIPv6 overlay tunnel includes:

- An EoL2TPv3oIPv6 overlay tunnel with and without VLAN tags:
 - Plain ethernet traffic OR
 - Dot1Q (Single VLAN tagged)
- All applicable features/functionality that are currently supported on physical interfaces for PPPoE sessions:
 - PTA (locally terminated) or LAC (forwarded to LNS over L2TPv2oIPv4)
 - IPv4 IPoE session (**Note:** IPv4 only)
 - Session authentication/authorization, policy enforcement, accounting and an AAA/RADIUS interface. These all function in the same way as currently supported on physical interfaces for PPPoE sessions
 - Only session-level QoS—as currently supported for PPPoE sessions.
 - For PPPoE PTA sessions:
 - Per-session in and out ACLs
 - VRF mapping

Overview of PPPoE over L2TPv3 Tunnels—Example Topology

The effect of using this feature can be simply shown by looking at example topologies before and after this feature was introduced. Figure 1 shows an example topology using a traditional BNG architecture. This example uses two BNGs for three CPEs. Figure 2 shows an example of the BNG architecture using this feature, which only uses one vBNG for three CPEs.

Figure 1: Traditional BNG Architecture

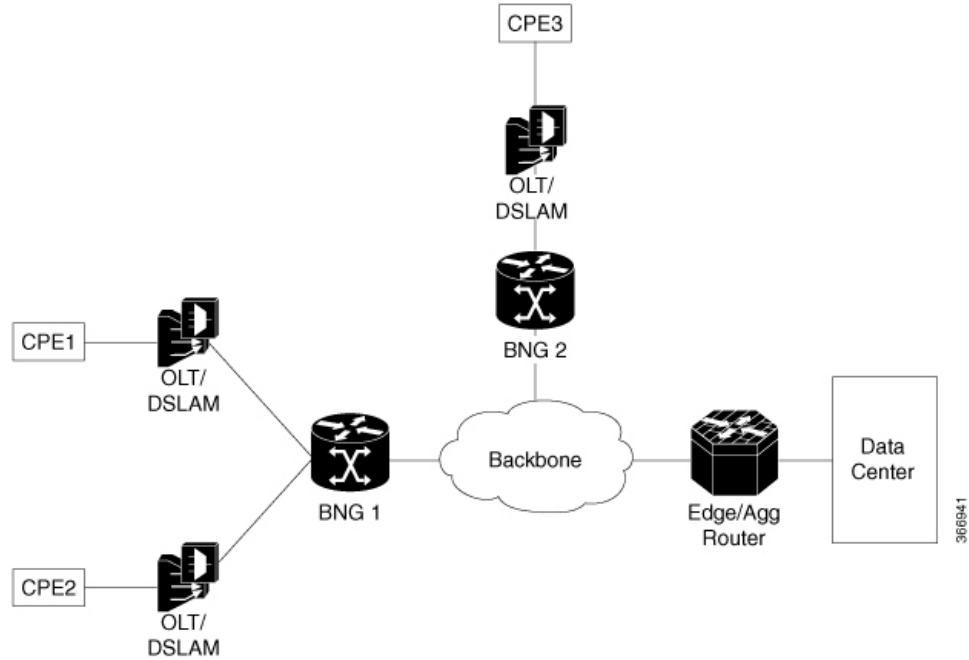
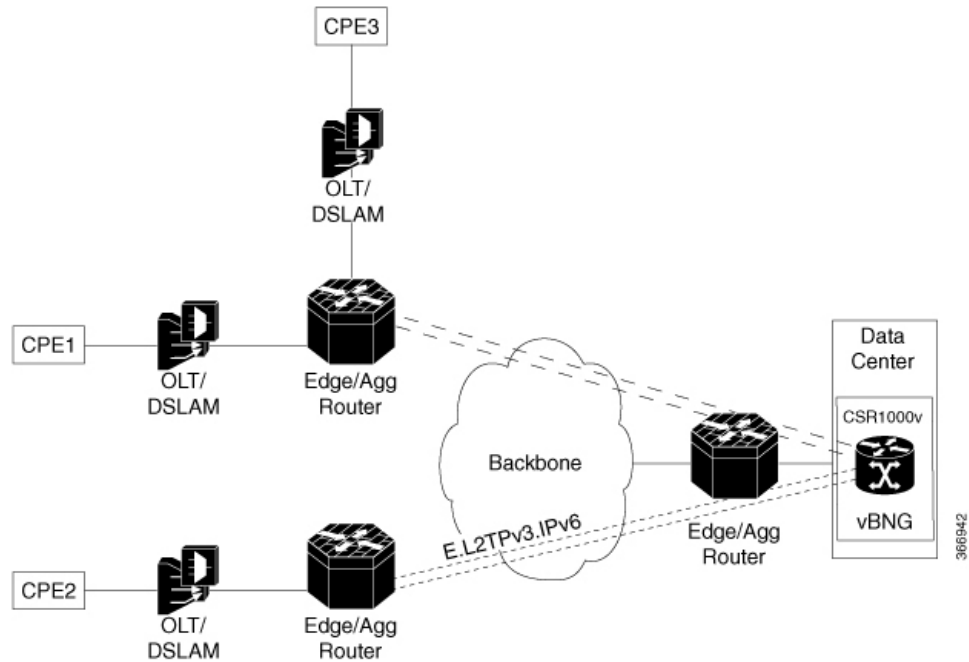


Figure 2: PPPoE over L2TPv3 Tunnels to vBNG



Benefits of PPPoE over L2TPv3 Tunneling

A benefit of this feature is that a Broadband Network Gateway(BNG) can be placed in each data center, instead of at each point of presence(PoP). An ISP can use L2TPv3 tunneling to send dual-stack PPP packets across

its own IPv6 backbone network for a PPP Terminated Aggregation (PTA) session or a L2TP Access Concentrator (LAC) session.

Prerequisites for PPPoE over L2TPv3 Tunnels

Software Prerequisites:

- Currently only supported on a Cisco CSR 1000v VM. This virtual router requires at least 2 CPUs, 8GB RAM or above, and 2 or more (vNIC) interfaces (10Gb).

Restrictions for Configuring PPPoE over L2TPv3 Tunnels

- Q-in-Q tunneling is not supported
- Tunnel H-QoS is not supported
- Access Node Control Protocol (ANCP) is not supported
- IPoE sessions are not supported (only PPPoE sessions are supported)
- Netconf/Yang Model is not supported
- We recommend using a physical interface/subinterface as the tunnel source instead of a loopback interface, to support session-level QoS Queuing or Shaping
- High Availability (HA) is not supported
- This feature does not support any PPPoE feature under the tunnel interface except for PPPoE enable function.
- A VLAN range under the tunnel is not supported
- MIB is not supported
- The size of the secondary local cookie must equal the size of the primary local cookie
- If a PPPoE session is up, the following actions are not allowed:
 - Removal of the tunnel mode
 - Removal of remote cookies
 - Modification or removal of tunnel parameters is not allowed, but removal of local cookies is allowed.

Scaling of L2TPv3 Tunnels

Performance of Cisco CSR 1000v

The scaling and throughput for vBNG on the Cisco CSR 1000v depends upon the compute Node platform and Operating System, including the hypervisor and vRouter. An example specification and the resulting performance, are described below,

Specification—An Intel x86 server platform consisting of a compute node running vBNG instances (Cisco CSR 1000v VMs) with 2 sockets (14 cores per socket), 4 x 10G NICs, a CPU of 2.30 GHz E5-2658 v4/105W

14C/35MB Cache/DDR4 2400MHz. A Linux Ubuntu 14.04 host OS with KVM Hypervisor (QEMU Rx & Tx size=1024) and a vSwitch (DPDK & vhost-user interface to the Cisco CSR 1000v VM). Note: We highly recommend using vCPU pinning for the Cisco CSR 1000v VMs and emulator, because large 1 GB pages are required for Cisco CSR 1000v VMs and the host OS.

Performance—8000 sessions (PPPoE) across 40 static/stateless P2P EoL2TPv3oIPv6 tunnels, with an average of 200 sessions per tunnel, a total throughput (UL + DL) of 4 Gbps. A vNIC with 2 x 10G ports: one port is for DL (to/from Edge/Aggregation router) and another port for UL (to/from core network).

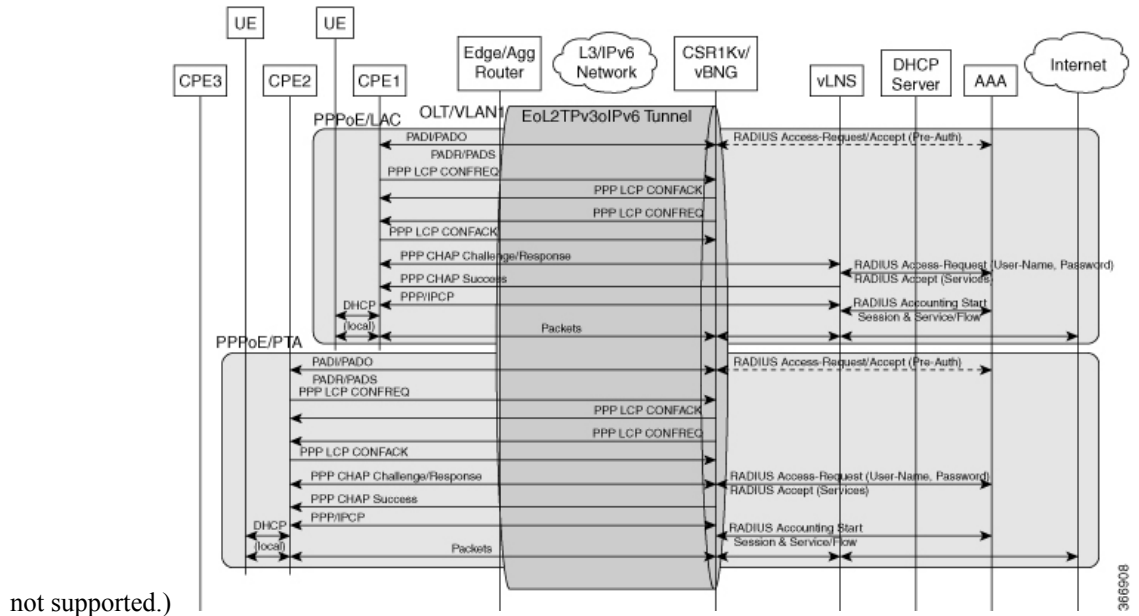
The following table shows the relationship between the number of tunnels and the number of PTA or LAC sessions per tunnel.

Table 1: Scaling for L2TPv3 Tunnels: Cisco CSR 1000v

No. of L2TPv3 over IPv6 Tunnels	PTA sessions per tunnel Single or Dual Stack	LAC sessions per tunnel Single or Dual Stack	PTA + LAC Sessions per tunnel Single or Dual Stack
40	200		
40		200	
40			200 PTA (in 30 tunnels) and 200 LAC (in 10 tunnels)

Call Flows for PPPoE over L2TPv3oIPv6 Tunnels

The figure below summarizes the call flows for PPPoE over an L2TPv3oIPv6 tunnel. Call flows are also explained here: [PPP and L2TP Flow Summary](#). For Cisco IOS XE Fuji 16.7, PPPoE is supported. (IPoE is



NAS-Port-Type Extensions

The following extended NAS-Port-Types are currently defined for a PPPoE service on ethernet and ATM interfaces.

- PPPoA—Radius value 30
- PPPoEoA—Radius value 31
- PPPoEoE—Radius value 3
- PPPoEoVLAN—Radius value 33
- PPPoEoQinQ—Radius value 34

In this feature, PPPoE support is added to the virtual interface (tunnel), which requires a new NAS-Port-Type for the PPPoE service on a virtual interface.

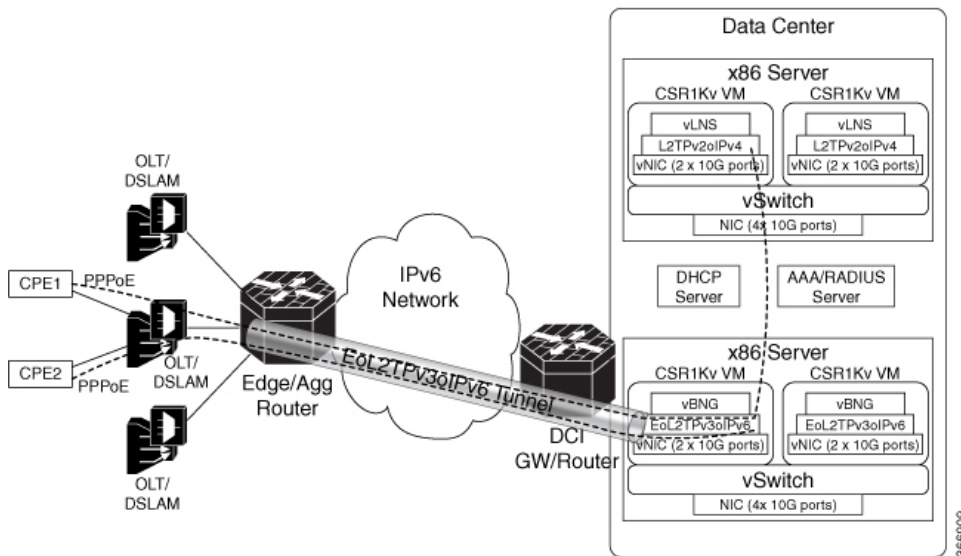
The following extended NAS-Port-Types were introduced for RFC2516, and which support the PPPoE service on virtual interfaces are supported:

- VirtualPPPoEoE (PPP over Ethernet [RFC2516] over Ethernet over tunnel/pseudowire) – Radius Value 44
- VirtualPPPoEoVLAN (PPP over Ethernet [RFC2516] over VLAN tunnel/pseudowire) – Radius Value 45

(The following extended NAS-Port-Type, introduced for RFC2516, is not supported: VirtualPPPoEoQinQ (PPP over Ethernet [RFC2516] over IEEE 802.1QinQ tunnel/pseudowire) – Radius Value 46.)

Network Topology Overview

Figure 3 below is an overview of the network topology. Traffic from CPE1 and CPE2 uses PPPoE sessions and EoL2TPv3oIPv6 tunnels to a vBNG in the data center. For each OLT/VLAN, a static EoL2TPv3oIPv6 tunnel is provisioned between an Edge or Aggregation Router and the vBNG on a Cisco CSR 1000v. The Edge/Agg Router forwards ethernet traffic from CPEs through EoL2TPv3oIPv6 tunnels.



How to Configure PPPoE over L2TPv3 Tunnels

Overview of Configuring PPPoE over L2TPv3 Tunnels

To configure PPPoE over L2TPv3 tunnels perform the following configuration tasks:

- [Configuring the Edge Router for PPPoE over L2TPv3 Tunneling](#)
- [Configuring the vSwitch, on page 10](#)
- [Verifying the vSwitch, on page 13](#)
- [Configuring Radius Authentication for PPPoE over L2TPv3 Tunnels, on page 15](#)
- [Configuring vBNG on the Cisco CSR 1000v VM](#)
- [Configuring PPPoE over an L2TPv3 Tunnel on the Cisco CSR 1000v](#)
- [Verifying PPPoE Sessions, on page 19](#)
- [Verifying PPPoE over L2TPv3 Tunnels, on page 21](#)
- [Troubleshooting PPPoE over L2TPv3 Tunnels, on page 27](#)

Configuring the Edge Router for PPPoE over L2TPv3 Tunneling

The following example shows the configuration of the edge router (for example, a Cisco ASR 9000) at the start of the L2TPv3 tunnel that extends from this router through the backbone network to an x86 server in the data center.

```
RP/0/RSP0/CPU0:ASR9K# show running-configuration

Thu Oct 19 02:04:51.459 UTC
Building configuration...
!! IOS XR Configuration 5.1.1.11I
!! Last configuration change at Thu Oct 12 15:29:01 2017 by lab
!
hostname ASR9K
logging console debugging
logging buffered 10000000
tftp vrf default ipv4 server homedir disk0:
cdp
cdp log adjacency changes
cdp advertise v1
line console
  timeout login response 0
  exec-timeout 0 0
  stopbits 1
  session-timeout 0
!
!
!
tftp client source-interface MgmtEth0/RSP0/CPU0/0
ipv4 access-list 1
  10 permit ipv4 30.1.1.0/24 any
!
```

```

interface MgmtEth0/RSP0/CPU0/0
  cdp
  ipv4 address 9.45.102.62 255.255.0.0
  !
interface MgmtEth0/RSP0/CPU0/1
  cdp
  shutdown
  !
interface TenGigE0/0/0/0
  ipv4 address 10.1.1.2 255.255.255.0
  transceiver permit pid all
  !
interface TenGigE0/0/0/0.401 l2transport
  encapsulation dot1q 401
  !
interface TenGigE0/0/0/0.402 l2transport
  encapsulation dot1q 402
  !
<for brevity, the interface commands from
interface TenGigE0/0/0/0.403 l2transport
to
interface TenGigE0/0/0/0.440 l2transport
have been removed>

interface TenGigE0/0/0/1
  ipv4 address 20.1.1.1 255.255.255.0
  ipv6 address 1111:2222::abcd/64
  ipv6 enable
  transceiver permit pid all
  !
interface TenGigE0/0/0/2
  ipv4 address 30.1.1.2 255.255.255.0
  !
interface TenGigE0/0/0/3
  ipv4 address 40.1.1.1 255.255.255.0
  !
interface preconfigure GigabitEthernet0/0/0/12
  ipv4 address 1.1.2.13 255.255.255.0
  shutdown
  transceiver permit pid all
  !
interface preconfigure GigabitEthernet0/0/0/13
  ipv4 address 1.1.2.14 255.255.255.0
  shutdown
  transceiver permit pid all
  !

<for brevity, some interface commands have been removed>

interface preconfigure GigabitEthernet0/0/0/19
  description test
  ipv4 address 2.2.2.5 255.255.255.0
  shutdown
  transceiver permit pid all
  !
interface preconfigure GigabitEthernet0/0/1/0
  ipv4 address 1.1.2.20 255.255.255.0
  shutdown
  transceiver permit pid all
  !
interface preconfigure GigabitEthernet0/0/1/1
  description test-pub-sub
  ipv4 address 1.1.2.21 255.255.255.0
  shutdown

```



```

    transceiver permit pid all
    !
interface preconfigure GigabitEthernet0/0/1/2
  ipv4 address 1.1.2.22 255.255.255.0
  shutdown
  transceiver permit pid all
  !

<for brevity, some interface commands have been removed>

interface preconfigure GigabitEthernet0/0/1/19
  ipv4 address 10.64.67.24 255.255.255.0
  shutdown
  transceiver permit pid all
  !
interface preconfigure GigabitEthernet0/1/0/0
  service-policy input ariadne
  service-policy output ariadne
  ipv4 address 1.1.1.1 255.255.255.0
  ipv4 access-group ariadne-demo-phase2 ingress
  !
interface preconfigure GigabitEthernet0/1/0/1
  ipv4 address 10.2.2.5 255.255.255.0
  shutdown
  !
interface preconfigure GigabitEthernet0/1/0/2
  ipv4 address 10.3.3.5 255.255.255.0
  !
interface preconfigure GigabitEthernet0/1/0/3
  ipv4 address 10.4.4.5 255.255.255.0
  !
interface preconfigure GigabitEthernet0/1/0/4
  shutdown
  !
<for brevity, the interface configuration commands from
interface preconfigure GigabitEthernet0/1/0/4
to
interface preconfigure GigabitEthernet0/2/1/19
have been removed>

shutdown
!
router static
  address-family ipv4 unicast
    1.1.0.0/16 30.1.1.1
    10.64.67.0/24 40.1.1.2
    50.1.1.0/24 40.1.1.2
    202.153.144.25/32 9.45.0.1
  !
  address-family ipv6 unicast
    1111:2222::cdef/128 TenGigE0/0/0/1
  !
!
l2vpn
  pw-class test
    encapsulation l2tpv3
    protocol l2tpv3
  !
!
xconnect group test1
  p2p test1
    interface TenGigE0/0/0/0.401
    neighbor ipv6 1111:2222::cdef pw-id 1

```

```

pw-class test1
source 1111:1101::abcd
l2tp static
local cookie size 8 value 0x4 0x4
local session 1
remote cookie size 8 value 0x4 0x4
remote session 1
!
!
!
<for brevity, the commands from
xconnect group test2
to
xconnect group test40
have been removed>

```

Configuring the vSwitch

To configure a Vector Packet Processing (VPP) vSwitch, which forwards L2TPv6 packets on the x86 server, edit the startup.conf file as shown in the following example. This allows the connection of physical NICs to vBNBs in the VMs.

```

root@vbnb:~# more /etc/vpp/startup.conf
unix {
  nodaemon
  cli-listen localhost:5002
  log /tmp/vpp.log
  full-coredump
}
cpu {
  ## In the VPP there is one main thread and optionally the user can create worker(s)

  ## The main thread and worker thread(s) can be pinned to CPU core(s) manually or
  automatically
  ## Manual pinning of thread(s) to CPU core(s)
  ## Set logical CPU core where main thread runs
  main-core 2
  ## Set logical CPU core(s) where worker threads are running
  corelist-workers 3,6
  ## Automatic pinning of thread(s) to CPU core(s)
  ## Sets number of CPU core(s) to be skipped (1 ... N-1)
  ## Skipped CPU core(s) are not used for pinning main thread and working thread(s).
  ## The main thread is automatically pinned to the first available CPU core and
  worker(s)
  ## are pinned to next free CPU core(s) after core assigned to main thread
  # skip-cores 4
  ## Specify a number of workers to be created
  ## Workers are pinned to N consecutive CPU cores while skipping "skip-cores" CPU
  core(s)
  ## and main thread's CPU core
  # workers 2
  ## Set scheduling policy and priority of main and worker threads
  ## Scheduling policy options are: other (SCHED_OTHER), batch (SCHED_BATCH)
  ## idle (SCHED_IDLE), fifo (SCHED_FIFO), rr (SCHED_RR)
  # scheduler-policy fifo
  ## Scheduling priority is used only for "real-time policies (fifo and rr),
  ## and has to be in the range of priorities supported for a particular policy
  # scheduler-priority 50
}
dpdk {

```

```

## Change default settings for all interfaces
# dev default {
    ## Number of receive queues, enables RSS
    ## Default is 1
    # num-rx-queues 3
    ## Number of transmit queues, Default is equal
    ## to number of worker threads or 1 if no workers threads
    # num-tx-queues 3
    ## Number of descriptors in transmit and receive rings
    ## increasing or reducing number can impact performance
    ## Default is 1024 for both rx and tx
    # num-rx-desc 512
    # num-tx-desc 512
    ## VLAN strip offload mode for interface
    ## Default is off
    # vlan-strip-offload on
# }
## Whitelist specific interface by specifying PCI address
# dev 0000:02:00.0
no-multi-seg
socket-mem 8192,8192
uio-driver igb_uio
dev 0000:43:00.0
dev 0000:43:00.1
## Whitelist specific interface by specifying PCI address and in
## addition specify custom parameters for this interface
# dev 0000:02:00.1 {
#     num-rx-queues 2
# }
## Change UIO driver used by VPP, Options are: igb_uio, vfio-pci
## and uio_pci_generic (default)
# uio-driver vfio-pci
## Disable multi-segment buffers, improves performance but
## disables Jumbo MTU support
# no-multi-seg
## Increase number of buffers allocated, needed only in scenarios with
## large number of interfaces and worker threads. Value is per CPU socket.
## Default is 16384
# num-mbufs 128000
## Change hugepages allocation per-socket, needed only if there is need for
## larger number of mbufs. Default is 256M on each detected CPU socket
# socket-mem 2048,2048
}
api-trace {
    on
}
api-segment {
    gid vpp
}
# Adjusting the plugin path depending on where the VPP plugins are:
#plugins
#{
#     path /home/bms/vpp/build-root/install-vpp-native/vpp/lib64/vpp_plugins
#}
# Alternate syntax to choose plugin path
#plugin_path /home/bms/vpp/build-root/install-vpp-native/vpp/lib64/vpp_plugins
root@vbng:~#

```

Telnet into VPP and check the interfaces:

```

root@gordian:~# telnet localhost 5002
Trying ::1...
Trying 127.0.0.1...
Connected to localhost.

```

Escape character is '^']'.

```
vpp#
vpp# show interfaces
```

Name	Idx	State	Counter	Count
TenGigabitEthernet48/0/0	1	down		
TenGigabitEthernet51/0/0	2	down		
local0	0	down		

```
vpp#
```

Configure VPP on the PPP to CSR 1000v side:

```
vpp# create vhost socket /tmp/vhost-user-vm1-int1
VirtualEthernet0/0/0
vpp#
vpp# set interface state VirtualEthernet0/0/0 up
vpp# set interface l2 bridge VirtualEthernet0/0/0 1

vpp# set interface state TenGigabitEthernet51/0/0 up
vpp# set interface l2 bridge TenGigabitEthernet51/0/0 2
vpp# create vhost socket /tmp/vhost-user-vm1-int2
VirtualEthernet0/0/1
vpp#
vpp# set interface state VirtualEthernet0/0/1 up
vpp# set interface l2 bridge VirtualEthernet0/0/1 2
```

Configure VPP on the backbone network side:

```
vpp# set interface state TenGigabitEthernet51/0/0 up
vpp# set interface l2 bridge TenGigabitEthernet51/0/0 2
vpp# create vhost socket /tmp/vhost-user-vm1-int2
VirtualEthernet0/0/1
vpp#
vpp# set interface state VirtualEthernet0/0/1 up
vpp# set interface l2 bridge VirtualEthernet0/0/1 2
```

Verify the vSwitch configuration using the `show interface` command as shown in the following example. This includes an incoming physical interface (pNIC) on the x86 server TenGigabitEthernet43/0/0. The corresponding virtual interface on the vSwitch is VirtualEthernet0/0/0.

```
vpp# show interface
```

Name	Idx	State	Counter	Count
TenGigabitEthernet43/0/0	1	up	rx packets	3750503475
			rx bytes	3999648912124
			tx packets	6441344996
			tx bytes	7212168652537
			drops	66263
TenGigabitEthernet43/0/1	2	up	rx packets	8058785942
			rx bytes	8503746413545
			tx packets	3074652529
			tx bytes	2909632140685
			tx-error	49
VirtualEthernet0/0/0	3	up	rx packets	6441344996
			rx bytes	7212168652537
			tx packets	3750437212
			tx bytes	3999640288634
			drops	251663
VirtualEthernet0/0/1	4	up	tx-error	66263
			rx packets	3074652578

```

rx bytes      2909632143625
tx packets    8058785942
tx bytes      8503746413545
drops         1460666

local0        0          down

vpp#

```

Verifying the vSwitch

Perform the following `show running-configuration` command to verify the configurations that were previously made in [Configuring the vSwitch, on page 10](#).

```
vpp# show running-configuration
```

```

Thread 0 vpp_main (lcore 2)
Time 1340274.6, average vectors/node 0.00, last 128 main loops 0.00 per node 0.00
  vector rates in 0.0000e0, out 0.0000e0, drop 0.0000e0, punt 0.0000e0
      Name              State      Calls      Vectors      Suspend
      Cllocks  Vectors/Call
acl-plugin-fa-cleaner-process  event wait      0          0          1
    3.83e4      0.00
admin-up-down-process          event wait      0          0          1
    1.60e4      0.00
api-rx-from-ring               active          0          0      68709
    1.82e4      0.00
bfd-process                     event wait      0          0          1
    2.97e4      0.00
cdp-process                     any wait      0          0      478747
    2.11e3      0.00
dhcp-client-process            any wait      0          0      13401
    1.13e4      0.00
dpdk-ipsec-process             done          1          0          0
    1.74e5      0.00
dpdk-process                   any wait      0          0      446669
    4.05e5      0.00
fib-walk                       any wait      0          0      670044
    4.35e3      0.00
flow-report-process            any wait      0          0          1
    2.52e4      0.00
flowprobe-timer-process        any wait      0          0          1
    6.17e4      0.00
gmon-process                   time wait      0          0      268018
    4.33e3      0.00
ikev2-manager-process          any wait      0          0      1340090
    4.09e3      0.00
ioam-export-process            any wait      0          0          1
    4.51e4      0.00
ip6-icmp-neighbor-discovery-ev any wait      0          0      1340090
    3.59e3      0.00
l2fib-mac-age-scanner-process  event wait      0          0          1
    1.51e4      0.00
lisp-retry-service             any wait      0          0      670044
    5.42e3      0.00
lldp-process                   event wait      0          0          1
    2.71e6      0.00
memif-process                  event wait      0          0          1
    2.59e5      0.00
nat64-expire-walk              done          1          0          0
    7.21e4      0.00
send-garp-na-process           event wait      0          0          1
    1.21e4      0.00
snat-det-expire-walk           done          1          0          0

```

```

      4.20e4      0.00
startup-config-process      done      1      0      1
      1.99e4      0.00
udp-ping-process      any wait      0      0      1
      7.34e4      0.00
unix-epoll-input      polling      2516699421      0      0
      1.17e6      0.00
vhost-user-process      any wait      0      0      446683
      1.09e4      0.00
vhost-user-send-interrupt-proc any wait      0      0      1
      1.16e4      0.00
vpe-link-state-process      event wait      0      0      63
      5.59e3      0.00
vpe-oam-process      any wait      0      0      656907
      3.59e3      0.00
vpe-route-resolver-process      any wait      0      0      13401
      1.09e4      0.00
vxlan-gpe-ioam-export-process any wait      0      0      1
      4.14e4      0.00
-----

```

Thread 1 vpp_wk_0 (lcore 3)

Time 1340274.6, average vectors/node 2.29, last 128 main loops 0.00 per node 0.00
vector rates in 7.6043e3, out 7.6042e3, drop 4.9439e-2, punt 0.0000e0

Name	State	Calls	Vectors	Suspends
TenGigabitEthernet43/0/0-output	active	1033884260	6441345157	0
2.48e1	6.23			
TenGigabitEthernet43/0/0-tx	active	1033884260	6441345157	0
1.07e2	6.23			
VirtualEthernet0/0/0-output	active	3477767073	3750503532	0
1.06e2	1.08			
VirtualEthernet0/0/0-tx	active	3477700818	3750437269	0
1.84e3	1.08			
dpdk-input	polling	16272823442465	3750503532	0
2.78e5	0.00			
error-drop	active	66255	66263	0
1.17e2	1.00			
ethernet-input	active	4429434865	10191848689	0
9.61e1	2.30			
l2-flood	active	81553	81594	0
5.29e2	1.00			
l2-fwd	active	4429356482	10191767095	0
6.80e1	2.30			
l2-input	active	4429434865	10191848689	0
1.55e2	2.30			
l2-learn	active	4429434865	10191848689	0
6.60e1	2.30			
l2-output	active	4429434865	10191848689	0
5.48e1	2.30			
vhost-user-input	polling	16252470346346	6441345157	0
2.26e5	0.00			

Thread 2 vpp_wk_1 (lcore 6)

Time 1340274.6, average vectors/node 1.34, last 128 main loops 0.00 per node 0.00
vector rates in 8.3068e3, out 8.3068e3, drop 3.6559e-5, punt 0.0000e0

Name	State	Calls	Vectors	Suspends
TenGigabitEthernet43/0/1-output	active	633497566	3074652580	0
3.73e1	4.85			
TenGigabitEthernet43/0/1-tx	active	633497517	3074652531	0
1.35e2	4.85			
VirtualEthernet0/0/1-output	active	7749969518	8058785944	0
1.23e2	1.04			
VirtualEthernet0/0/1-tx	active	7749969518	8058785944	0

```

1.69e3          1.04
dpdk-input     1.32e5          0.00      polling  15572336805744      8058785944      0
1.32e5          0.00
error-drop     6.62e2          1.00      active   49                49                0
6.62e2          1.00
ethernet-input 1.17e2          1.34      active   8294784335        11133438524      0
1.17e2          1.34
l2-flood       7.26e2          1.00      active   142               142               0
7.26e2          1.00
l2-fw          9.45e1          1.34      active   8294784194        11133438382      0
9.45e1          1.34
l2-input       2.81e2          1.34      active   8294784335        11133438524      0
2.81e2          1.34
l2-learn       1.65e2          1.34      active   8294784335        11133438524      0
1.65e2          1.34
l2-output      8.25e1          1.34      active   8294784335        11133438524      0
8.25e1          1.34
vhost-user-input 4.65e5          0.00      polling  15552267493917    3074652580      0
4.65e5          0.00

vpp#

```

Configuring Radius Authentication for PPPoE over L2TPv3 Tunnels

Configuring Radius Authentication for a PPP PTA Session

To configure Radius authentication for a PPP terminated aggregation (PTA) session, refer to a guide such as the [Radius Configuration guide](#). The following example shows a snippet of the code required for the authentication of the PPP PTA session.

```

pta@cisco.com Cleartext-Password := "cisco"
                Service-Type = Framed-user,
#               Cisco-Account-Info += "Asrl_down(r=500)",
#               Cisco-Account-Info += "Asrl_up(r=500,corr=20,si=1)",
                Cisco-Account-Info += "Asrl_down(r=200)",
#               Session-Timeout = 400,
                Cisco-Account-Info += "subscriber:accounting-list=List1",
                Cisco-Account-Info += "AACCT_SERVICE",
                Cisco-Account-Info += "AACCT_SERVICE_V6"

```

Configuring Radius Authentication for a PPP LAC Session

To configure Radius authentication for the L2TP Access Concentrator (LAC), refer to a RADIUS guide such as the [Radius Configuration guide](#). The following example shows a snippet of the code required for authentication of the LAC.

```

lacupsrl@cisco.com Cleartext-Password := "cisco"
                    Service-Type = Outbound-User,
                    cisco-AVPair += "vpdn:tunnel-id=CSR_TENGIG",
                    cisco-AVPair += "vpdn:l2tp-tunnel-password=cisco",
                    cisco-AVPair += "vpdn:tunnel-type=l2tp",
                    cisco-AVPair += "vpdn:ip-addresses=40.1.1.2",
#                   Session-Timeout = 600,
                    Cisco-Account-Info += "Asrl_up(r=500,corr=20,si=1)",
                    Cisco-AVPair += "ip:outacl=acct_out",
                    Cisco-Account-Info += "subscriber:accounting-list=List1"

```

Configuring vBNG on the Cisco CSR 1000v VM

Configure the XML configuration file for vBNG on the Cisco CSR 1000v VM as shown in the following example.

```
<domain type='kvm' id='21' xmlns:qemu='http://libvirt.org/schemas/domain/qemu/1.0'>
  <name>vbng</name>
  <uuid>7c0c20b3-b9b6-462c-a1e6-01d3efac0abe</uuid>
  <memory unit='KiB'>8388608</memory>
  <currentMemory unit='KiB'>8388608</currentMemory>
  <memoryBacking>
    <hugepages/>
  </memoryBacking>
  <vcpu placement='static'>28</vcpu>
  <os>
    <type arch='x86_64' machine='pc-i440fx-xenial'>hvm</type>
    <boot dev='hd'/>
  </os>
  <features>
    <acpi/>
    <apic/>
  </features>
  <cpu mode='host-model'>
    <model fallback='allow'/>
    <topology sockets='2' cores='14' threads='1'/>
  </cpu>
  <clock offset='utc'>
    <timer name='rtc' tickpolicy='catchup'/>
    <timer name='pit' tickpolicy='delay'/>
    <timer name='hpet' present='no'/>
  </clock>
  <on_poweroff>destroy</on_poweroff>
  <on_reboot>restart</on_reboot>
  <on_crash>restart</on_crash>
  <pm>
    <suspend-to-mem enabled='no'/>
    <suspend-to-disk enabled='no'/>
  </pm>
  <devices>
    <emulator>/usr/bin/kvm-spice</emulator>
    <disk type='file' device='disk'>
      <driver name='qemu' type='qcow2'/>
      <source file='/var/lib/libvirt/images/vbng.qcow2'/>
      <backingStore/>
      <target dev='hda' bus='ide'/>
      <alias name='ide0-0-0'/>
      <address type='drive' controller='0' bus='0' target='0' unit='0'/>
    </disk>
    <disk type='file' device='cdrom'>
      <driver name='qemu' type='raw'/>
      <backingStore/>
      <target dev='hdb' bus='ide'/>
      <readonly/>
      <alias name='ide0-0-1'/>
      <address type='drive' controller='0' bus='0' target='0' unit='1'/>
    </disk>
    <controller type='usb' index='0' model='ich9-ehci1'>
      <alias name='usb'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x06' function='0x7'/>
    </controller>
    <controller type='usb' index='0' model='ich9-uhci1'>
      <alias name='usb'/>
    </controller>
  </devices>
</domain>
```



```

    <master startport='0'>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x06' function='0x0'
multifunction='on'>/>
    </controller>
    <controller type='usb' index='0' model='ich9-uhci2'>
      <alias name='usb'>/>
      <master startport='2'>
        <address type='pci' domain='0x0000' bus='0x00' slot='0x06' function='0x1'>/>
      </controller>
    <controller type='usb' index='0' model='ich9-uhci3'>
      <alias name='usb'>/>
      <master startport='4'>
        <address type='pci' domain='0x0000' bus='0x00' slot='0x06' function='0x2'>/>
      </controller>
    <controller type='pci' index='0' model='pci-root'>
      <alias name='pci.0'>/>
    </controller>
    <controller type='ide' index='0'>
      <alias name='ide'>/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x01' function='0x1'>/>
    </controller>
    <controller type='virtio-serial' index='0'>
      <alias name='virtio-serial0'>/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x05' function='0x0'>/>
    </controller>
    <interface type='network'>
      <mac address='52:54:00:80:e7:05'>/>
      <source network='default' bridge='virbr0'>/>
      <target dev='vnet0'>/>
      <model type='rtl8139'>/>
      <alias name='net0'>/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x03' function='0x0'>/>
    </interface>
    <serial type='pty'>
      <source path='/dev/pts/5'>/>
      <target port='0'>/>
      <alias name='serial0'>/>
    </serial>
    <console type='pty' tty='/dev/pts/5'>
      <source path='/dev/pts/5'>/>
      <target type='serial' port='0'>/>
      <alias name='serial0'>/>
    </console>
    <channel type='spicevmc'>
      <target type='virtio' name='com.redhat.spice.0' state='disconnected'>/>
      <alias name='channel0'>/>
      <address type='virtio-serial' controller='0' bus='0' port='1'>/>
    </channel>
    <redirdev bus='usb' type='spicevmc'>
      <alias name='redir0'>/>
    </redirdev>
    <redirdev bus='usb' type='spicevmc'>
      <alias name='redir1'>/>
    </redirdev>
    <memballoon model='virtio'>
      <alias name='balloon0'>/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x07' function='0x0'>/>
    </memballoon>
  </devices>
  <seclabel type='dynamic' model='apparmor' relabel='yes'>
    <label>libvirt-7c0c20b3-b9b6-462c-a1e6-01d3efac0abe</label>
    <imagelabel>libvirt-7c0c20b3-b9b6-462c-a1e6-01d3efac0abe</imagelabel>
  </seclabel>
  <qemu:commandline>

```

```

    <qemu:arg value='-numa' />
    <qemu:arg value='node,memdev=mem' />
    <qemu:arg value='-mem-prealloc' />
    <qemu:arg value='-object' />
    <qemu:arg
value='memory-backend-file,id=mem,size=8G,mem-path=/dev/hugepages,share=on,share=on' />
    <qemu:arg value='-netdev' />
    <qemu:arg value='vhost-user,id=hostnet1,chardev=vhost-user-vm1-int,vhostforce' />
    <qemu:arg value='-device' />
    <qemu:arg
value='virtio-net-pci,netdev=hostnet1,id=net1,mac=52:54:00:00:01:01,mrg_rxbuf=on' />
    <qemu:arg value='-chardev' />
    <qemu:arg value='socket,id=vhost-user-vm1-int,server,path=/tmp/vhost-user-vm1-int1' />
    <qemu:arg value='-netdev' />
    <qemu:arg value='vhost-user,id=hostnet2,chardev=vhost-user-vm1-int2,vhostforce' />
    <qemu:arg value='-device' />
    <qemu:arg
value='virtio-net-pci,netdev=hostnet2,id=net2,mac=52:54:00:00:01:02,mrg_rxbuf=on' />
    <qemu:arg value='-chardev' />
    <qemu:arg value='socket,id=vhost-user-vm1-int2,server,path=/tmp/vhost-user-vm1-int2' />
  </qemu:commandline>
</domain>

```

Configuring PPPoE over an L2TPv3 Tunnel on the Cisco CSR 1000v

Enter the following commands on the Cisco CSR 1000v, to configure PPPoE over an L2TPv3 tunnel.

```

configure terminal
interface Tunnel0
  mac-address 0000.5e00.5213
  ip address 10.10.151.1 255.255.255.0
  tunnel mode ethernet l2tpv3 manual
  l2tp id 222 111
  l2tp cookie local 8 54321
  l2tp cookie remote 8 12345
  tunnel destination 2001:DB8:1111:2222::1
  tunnel source 2001:DB8:2:2::1
  tunnel VLAN 10
  service-policy type control POLICY1
  pppoe enable group 1

```

`tunnel mode ethernet l2tpv3` specifies L2TPv3 as the tunneling method. The `l2tp id` command specifies the local and remote session IDs and the `l2tp cookie local` command configures a cookie field with a size in bytes (for example, 8 bytes) and a low value (for example, 54321).

`l2tp cookie remote` configures a cookie field with a size in bytes (for example, 8 bytes) and a low value (for example, 12345). The cookie field is part of the Layer 2 Tunnel Protocol Version 3 (L2TPv3) headers in outgoing packets that are sent from the local PE peer router.

The `tunnel destination` is the destination IPv6 address on the OLT/DSLAM that connects to the CPE. This is the end tunnel destination from the point of view of the vBNG (on the Cisco CSR 1000v).

`tunnel source` —source IPv6 address of the vBNG (on the Cisco CSR 1000v, on the x86 server).

This is another example showing different cookie options:

```

interface Tunnel1
  mac-address aaaa.bbbb.1101

```

```

no ip address
ip verify unicast source reachable-via rx
pppoe enable group global
tunnel source GigabitEthernet1
tunnel mode ethernet l2tpv3 manual
tunnel destination 1111:1101::ABCD
tunnel path-mtu-discovery
tunnel vlan 401
l2tp static remote session 1
l2tp static remote cookie size 8 value 0x4 0x4
l2tp static local session 1
l2tp static local cookie size 8 value 0x4 0x4
!
```

Refer to the Cisco CSR 1000v router configurations in [Configuring PPPoE over an L2TPv3 Tunnel on the Cisco CSR 1000v](#), on page 18.

Verifying PPPoE Sessions

To display information about PPPoE sessions running on the L2TPv3 tunnels, enter the **show pppoe summary** command.

This displays a summary of the currently active PPP over Ethernet (PPPoE) sessions per interface. For further information, see the [Cisco IOS Broadband Access Aggregation and DSL Command Reference](#). The following example shows summary information about the PPPoE sessions running over tunnel Tunnell.

```

Device# show pppoe summary
Load for five secs: 0%/0%; one minute: 0%; five minutes: 0%
No time source, *10:26:27.815 UTC Thu Oct 19 2017

PTA : Locally terminated sessions
FWDED: Forwarded sessions
TRANS: All other sessions (in transient state)

                TOTAL      PTA      FWDED      TRANS
TOTAL                1         1         0         0
Tunnell              1         1         0         0
Device#
```

Enter the **show pppoe session** command to display the currently active PPP over Ethernet (PPPoE) sessions.

```

Device# show pppoe session
Load for five secs: 15%/0%; one minute: 14%; five minutes: 13%
No time source, *16:19:38.030 UTC Thu Oct 26 2017

8000 sessions in LOCALLY_TERMINATED (PTA) State
8000 sessions total

Uniq ID  PPPoE  RemMAC      Port      VT  VA      State
      SID  LocMAC
      596   596  0011.9400.0080 Tu1      1  Vi2.585  PTA
      aaaa.bbbb.1101
      598   598  0011.9400.0081 Tu1      1  Vi2.586  PTA
      aaaa.bbbb.1101
      601   601  0011.9400.0082 Tu1      1  Vi2.592  PTA
      aaaa.bbbb.1101
      605   605  0011.9400.0083 Tu1      1  Vi2.593  PTA
      aaaa.bbbb.1101
      608   608  0011.9400.0084 Tu1      1  Vi2.600  PTA
      aaaa.bbbb.1101
      612   612  0011.9400.0085 Tu1      1  Vi2.602  PTA
```

```

          aaaa.bbbb.1101                                UP
616      616 0011.9400.0086 Tu1                        1 Vi2.606   PTA
          aaaa.bbbb.1101                                UP
619      619 0011.9400.0087 Tu1                        1 Vi2.610   PTA
          aaaa.bbbb.1101                                UP
1        1   0011.9400.0008 Tu1                        1 Vi2.3     PTA

```

Device#

Perform the `show platform hardware qfp active feature subscriber segment` command to view the segment ids. Example:

```

Device# show platform hardware qfp active feature subscriber segment
Load for five secs: 12%/0%; one minute: 13%; five minutes: 13%
No time source, *16:22:02.582 UTC Thu Oct 26 2017

```

Current number segments: 16000

Segment	SegType	QFP Hdl	PeerSeg	Status
0x0000003600003006	PPPOE	49	0x000000360000600b	BOUND
0x000000360000600b	LTERM	49	0x0000003600003006	BOUND
0x0000003800004008	PPPOE	50	0x000000380000700c	BOUND
...				

Device#

To display the PPPoE encapsulation, enter the `show platform software subscriber fp active segment id` command as shown in the following example. This example shows the PPPoE encapsulation for one of the segments identified in the previous example, with Segment ID= 0x0000003600003006.

```

Device# show platform software subscriber fp active segment id 0x0000003600003006

```

```

Load for five secs: 14%/0%; one minute: 13%; five minutes: 13%
No time source, *16:24:11.425 UTC Thu Oct 26 2017

```

Segment	SegType	EVSI	Changes	AOM Id	AOM State
0x0000003600003006	PPPoE	54	0x00000000	651	created

PPPoE Session id 0x3

MAC enctype 0x1a

Switch Mode 0x2

Max MTU 0x5b4

VLAN cos 0x8

Phy Intf (on CPP) 0x10

Conditional Debug OFF

MAC Address Local: aaaabbbb1109

MAC Address Remote: 002194000008

PPPoE encap string [76

bytes]: 6000000000073ff111122220000000000000000000000cdef1111110900000000000000000000
abcd00000090000000400000004002194000008aaaabbbb1109810001998864110000030000

Flow Information:

Flows activated/attached: 2/2

Input Classes: 2

Id	Priority	Flow EVSI	Class-Group Id	Filter Type:	Filter Name
2	0	4210704	368259939.1	Named ACL:	acct_in
4	0	4210705	368259939.2	Named ACL:	acct_inv6

```

Output Classes: 2
  Id      Priority  Flow EVSI  Class-Group Id  Filter Type:
                                     Filter Name
-----
          3         0    4210704    146674515.1    Named ACL:
                                     acct_out
          5         0    4210705    146674515.2    Named ACL:
                                     acct_outv6
    
```

Device#

To display MAC string information for a segment, use the `show platform hardware qfp` command as shown in the following example:

```
Device# show platform hardware qfp active feature subscriber segment id 0x0000003600003006
```

```

Load for five secs: 12%/0%; one minute: 13%; five minutes: 13%
No time source, *16:28:59.639 UTC Thu Oct 26 2017
Segment ID: 0x3600003006
    
```

```

EVSI: 54
Peer Segment ID: 0x360000600b
QFP vsi if handle: 49
QFP interface name: EVSI54
Segment type: PPPOE
Is conditional debug: 0
Is SIP: 1
Segment status: BOUND
Macstring length: 76
00000000 6000 0000 0000 73ff 1111 2222 0000 0000
00000010 0000 0000 0000 cdef 1111 1109 0000 0000
00000020 0000 0000 0000 abcd 0000 0009 0000 0004
00000030 0000 0004 0021 9400 0008 aaaa bbbb 1109
00000040 8100 0199 8864 1100 0003 0000
    
```

```

Encap info exmem handle: 0x0
session id: 3
vcd: 409
mtu: 1460
physical if handle: 16
hash value: 0x0000bd8c
Input Classes: 2
  Class Id  Flow EVSI      CG Id              QFP Hdl
-----
          2    4210704    368259939.1        54
          4    4210705    368259939.2        55
Output Classes: 2
  Class Id  Flow EVSI      CG Id              QFP Hdl
-----
          3    4210704    146674515.1        54
          5    4210705    146674515.2        55
    
```

Device#

Verifying PPPoE over L2TPv3 Tunnels

To show the adjacency created after a PPPoE session is enabled for an L2TPv3 tunnel (Tunnel1), enter the `show adjacency encapsulation` and `show adjacency detail` commands on the Cisco CSR 1000v. The

PPPoE session goes from the vBNG on the Cisco CSR 1000v VM to the Agg/Edge router (for example, a Cisco ASR 9000). The following examples

```
vBNG# show adjacency encapsulation
```

```
RAW      Tunnel1                               point2point(3)
Encap length 70
60000000000073FF1111222200000000
000000000000CDEF1111110100000000
000000000000ABCD0000000100000004
00000004000000000000AAAABBBB1101
810001918864
Provider: TUNNEL
| Protocol header count in macstring: 1
  HDR 0: ipv6
        dst: static, 1111:1101::ABCD
        src: static, 1111:2222::CDEF
        prot: static, 115
        tc: static, 0
        flow: static, 0
        hops: static, 255
        per packet fields: payload_length
```

```
vBNG# show adjacency detail
```

```
IPV6      GigabitEthernet1                     FE80::AA0C:DFF:FE53:2061(3)
                                                0 packets, 0 bytes
                                                epoch 0
                                                sourced in sev-epoch 45988
                                                Encap length 14
                                                A80C0D53206152540000010186DD
                                                L2 destination address byte offset 0
                                                L2 destination address byte length 6
                                                Link-type after encap: ipv6
                                                IPv6 ND
```

Show the running configuration for the L2TPv3 tunnel interface.

```
vBNG# show running-configuration interface Tunnel1
```

```
Load for five secs: 1%/0%; one minute: 0%; five minutes: 0%
No time source, *10:58:51.169 UTC Wed Oct 18 2017
```

```
Building configuration...
```

```
Current configuration : 440 bytes
!
interface Tunnel1
 mac-address aaaa.bbbb.1101
 no ip address
 ip verify unicast source reachable-via rx
 pppoe enable group global
 tunnel source GigabitEthernet1
 tunnel mode ethernet l2tpv3 manual
 tunnel destination 1111:1101::ABCD
 tunnel path-mtu-discovery
 tunnel vlan 401
 l2tp static remote session 1
 l2tp static remote cookie size 8 value 0x4 0x4
 l2tp static local session 1
 l2tp static local cookie size 8 value 0x4 0x4
```

To show the L2TPv3 tunnel interface, enter the `show interface` command as shown in the following example.

```
vBNG# show interface Tunnel1
```

```

Load for five secs: 0%/0%; one minute: 0%; five minutes: 0%
No time source, *10:59:18.610 UTC Wed Oct 18 2017

Tunnell is up, line protocol is up
  Hardware is Tunnel
  MTU 1460 bytes, BW 100 Kbit/sec, DLY 50000 usec,
    reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation TUNNEL, loopback not set
  Keepalive not set
  Tunnel linestate evaluation up
  Tunnel source 1111:2222::CDEF (GigabitEthernet1), destination 1111:1101::ABCD
  Tunnel Subblocks:
    src-track:
      Tunnell source tracking subblock associated with GigabitEthernet1
      Set of tunnels with source GigabitEthernet1, 41 members (includes iterators), on
interface <OK>
  Tunnel protocol/transport L2TP/IPV6
    L2TPv3
      remote session-id:1
      local session-id:1
      local cookie size:8, low value:0x4, high value:0x4
      remote cookie size:8, low value:0x4, high value:0x4
  Tunnel TTL 255
  Path MTU Discovery, ager 10 mins, min MTU 1280
  Tunnel transport MTU 1460 bytes
  Tunnel transmit bandwidth 8000 (kbps)
  Tunnel receive bandwidth 8000 (kbps)
  Last input never, output 1d03h, output hang never
  Last clearing of "show interface" counters 3d03h
  Input queue: 0/375/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/0 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
    9583487 packets input, 9983601325 bytes, 0 no buffer
    Received 0 broadcasts (0 IP multicasts)
    0 runts, 0 giants, 0 throttles
    0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
    13458 packets output, 1739662 bytes, 0 underruns
    0 output errors, 0 collisions, 0 interface resets
    0 unknown protocol drops
    0 output buffer failures, 0 output buffers swapped out
vBNG#

```

The following command shows the IPsec Cisco Quantum Flow Processor (QFP) information about tunnel1.

```
vBNG# show platform hardware qfp active feature tunnel interface tunnel1
```

```

Load for five secs: 0%/0%; one minute: 0%; five minutes: 0%
No time source, *10:56:51.924 UTC Wed Oct 18 2017
General interface info:
  Interface name: Tunnell
  Platform interface handle: 11
  QFP interface handle: 8
  QFP complex: 0
  Rx uIDB: 65530 Tx uIDB: 65528
  Hash index : 0x0003f6
  Hash element ppe addr : 0xe89bf440
  ESP Hash element ppe addr : 00000000
  AH Hash element ppe addr : 00000000
  UDP Hash element ppe addr : 00000000
  Output sb ppe addr : 0xe816fc20
  Decap chk sb ppe addr : 00000000
  DMVPN sb ppe addr input: 00000000 output: 00000000
  SGRE input sb ppe addr : 00000000

```

```

L2TPOIPV6 input sb ppe addr : 0xe75c6000
Config:
mode: L2TPV3oIPV6
src IP: 1111:2222:0000:0000:0000:0000:0000:cdef dest IP:
1111:1101:0000:0000:0000:0000:0000:abcd
ipv4_intf_vrf: 0 tun_vrf: 0 tun_vrf_egress: 0
key: 0 flags: 0x0081 app_id: TUN_APP_CLI app_data: 0
ttl: 255 tos: 0
tunnel_protection: FALSE
virtual MAC: aaaa.bbbb.1101
lport: 0 rport: 0
tunnel_enable_entropy: FALSE
remote_session_id: 1 vlan id for l2tpoipv6: 401
remote_cookie_size: 8 local_cookie_size: 8 local_cookie_secondary_size: 255
remote_cookie_low: 4 remote_cookie_high: 4
local_cookie_low: 4 local_cookie_high: 4
local_cookie_secondary_low: 0 local_cookie_secondary_high: 0

```

The following show running-configuration command, shows the Cisco CSR 1000v VM configuration on the x86 server.

```

vBNG# show running-configuration

Building configuration...

Current configuration : 4401 bytes
!
! Last configuration change at 13:38:48 IST Tue Oct 17 2017
!
version 16.6
service timestamps debug datetime msec
service timestamps log datetime msec
platform qfp utilization monitor load 80
no platform punt-keepalive disable-kernel-core
!
hostname vBNG
!
boot-start-marker
boot system harddisk:asr1000rpx86-universalk9.2017-06-16_14.49_vijasin3.SSA.bin
boot-end-marker
!
!
vrf definition Mgmt-intf
!
address-family ipv4
exit-address-family
!
address-family ipv6
exit-address-family
!
logging buffered 10000000
no logging console
!
aaa new-model
!
!
aaa group server radius GROUP1
server name RAD1
!
aaa group server radius acct
server name RAD1
!

```



```
aaa authentication login CONSOLE none
aaa authentication ppp default local
aaa authorization network default local
aaa authorization subscriber-service default local
aaa accounting update periodic 1
aaa accounting network default start-stop group GROUP1
aaa accounting network List1 start-stop group GROUP1
!
aaa session-id common
clock timezone IST 5 30
!
subscriber templating
subscriber accounting accuracy 10000
!
ipv6 unicast-routing
!
!
multilink bundle-name authenticated
vpdn enable
vpdn authen-before-forward
!
vpdn-group test1
  accept-dialin
  protocol l2tp
  virtual-template 1
  terminate-from hostname CSR_TENGIG
  l2tp tunnel password 0 cisco
!
!
crypto pki trustpoint TP-self-signed-336246438
  enrollment selfsigned
  subject-name cn=IOS-Self-Signed-Certificate-336246438
  revocation-check none
  rsakeypair TP-self-signed-336246438
!
!
crypto pki certificate chain TP-self-signed-336246438
!
!
license udi pid ASR1006 sn NWG123704SS
license accept end user agreement
license boot level advenenterprise
spanning-tree extend system-id
diagnostic bootup level minimal
!
!
username user1@cisco.com password 0 cisco
username lac@cisco.com password 0 cisco
username user1lac@cisco.com password 0 cisco
username user2lac@cisco.com password 0 cisco
username lacupsrl@cisco.com password 0 cisco
username lacdownsrl@cisco.com password 0 cisco
!
redundancy
  mode sso
!
!
interface GigabitEthernet0/0/0
  ip address 9.45.102.50 255.255.0.0
  ip nat outside
  negotiation auto
!
<commands removed for brevity>
```

```

interface TenGigabitEthernet0/2/0
 ip address 40.1.1.2 255.255.255.0
 ip nat inside
!
interface TenGigabitEthernet0/3/0
 ip address 50.1.1.1 255.255.255.0
!
interface GigabitEthernet0
 vrf forwarding Mgmt-intf
 no ip address
 negotiation auto
!
interface Virtual-Template1
 ip unnumbered TenGigabitEthernet0/2/0
 peer default ip address pool test1
 peer default ipv6 pool test1_v6
 ipv6 unnumbered TenGigabitEthernet0/2/0
 ppp mtu adaptive
 ppp authentication pap chap
!
interface Virtual-Template2
 ip unnumbered GigabitEthernet0/0/3
 ppp authentication pap
!
ip local pool test1 2.1.1.1 2.1.1.254
ip nat inside source list 1 interface GigabitEthernet0/0/0 overload
ip forward-protocol nd
ip http server
ip http authentication local
ip http secure-server
ip tftp source-interface GigabitEthernet0/0/0
ip tftp blocksize 8192
ip route 0.0.0.0 0.0.0.0 9.45.0.1
ip route 1.1.0.0 255.255.0.0 40.1.1.1
ip route 10.1.1.0 255.255.255.0 40.1.1.1
ip route 20.1.1.0 255.255.255.0 40.1.1.1
ip route 30.1.1.0 255.255.255.0 40.1.1.1
ip route 100.1.0.0 255.255.0.0 60.1.1.1
ip route 200.1.0.0 255.255.0.0 60.1.1.1
ip route 202.153.144.25 255.255.255.255 9.45.0.1
!
ip ssh server algorithm encryption aes128-ctr aes192-ctr aes256-ctr
ip ssh client algorithm encryption aes128-ctr aes192-ctr aes256-ctr
!
access-list 1 permit 30.1.1.0 0.0.0.255
access-list 1 permit any
ipv6 local pool test1_v6 5555::/48 64
!
radius server RAD1
 address ipv4 10.64.67.97 auth-port 1645 acct-port 1646
 key cisco123
!
!
control-plane
!
line con 0
 stopbits 1
line vty 0 4
!
end

```

Troubleshooting PPPoE over L2TPv3 Tunnels

Enter the following command to obtain information about a tunnel:

```
debug tunnel l2tp ipv6
```

In the following example, the normal case is shown, where the tunnel interface is shown as being up and running.

```
vBNG# debug tunnel l2tp ipv6
*Oct 19 10:32:38.475: %LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel130, changed
state to up
*Oct 19 10:32:38.478: Tunnel130: Adding 12 bytes for l2tp header
*Oct 19 10:32:38.478: Tunnel130: L2TPv3 header session id: 0x1E, cookie low: 0x4, cookie
high: 0x4
*Oct 19 10:32:38.478: Tunnel130: Adding 18 bytes for ethernet header
*Oct 19 10:32:38.478: Tunnel130: Ethernet header, dst mac:0000.0000.0000, src
mac:aaaa.bbbb.1130, dot1q, vlan:430, ethertype:34916
*Oct 19 10:32:38.485: %LINK-3-UPDOWN: Interface Tunnel130, changed state to up
```

Additional References for Configuring PPPoE over L2TPv3 Tunnels

Related Documents

Related Topic	Document Title
PPP over Ethernet	Broadband Access Aggregation and DSL Configuration Guide
Virtual Private Dialup Networks (VPDNs)	Virtual Private Dialup Network (VPDN)
Configuring VPDNs	VPDN Configuration Guide
Configuring L2TPv3 over IPv6Tunnels	L2VPN and Ethernet Services Configuration Guide
L2TP VPDN tunnels	VPDN Tunnel Management

RFCs

RFCs	Title
RFC 8159	Keyed IPv6 Tunnel

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Configuring PPPoE over L2TPv3 Tunnels

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 2: Feature Information for Configuring PPPoE over L2TPv3 Tunnels

Feature Name	Software Releases	Feature Information
Configuring PPPoE over L2TPv3 Tunnels	Cisco IOS XE Fuji 16.7.1	Use this feature to transport PPPoE packets between network sites in Layer 2 Tunneling Protocol Version 3 (L2TPv3) tunnels using IPv6.