

L2TP Subscriber Management

- Feature Summary and Revision History, on page 1
- Feature Description, on page 2
- L2TP tunnel failover support, on page 4
- Seamless L2TP session continuity during server reboot, on page 7
- IETF Tagged Attributes on LAC, on page 8
- How it Works, on page 13
- Configuring the L2TP Subscriber Management Feature, on page 23
- Monitoring and troubleshooting, on page 26

Feature Summary and Revision History

Summary Data

Table 1: Summary Data

Applicable Product(s) or Functional Area	cnBNG
Applicable Platform(s)	SMI
Feature Default Setting	Disabled - Configuration Required
Related Changes in this Release	Not Applicable
Related Documentation	Not Applicable

Revision History

Table 2: Revision History

Revision Details	Release
Introduced support for L2TP tunnel reboot.	2026.01.0

Revision Details	Release
Introduced the **show l2tp-tunnel-instance upf** command for viewing L2TP tunnel instance details.	2026.01.0
Introduced the **show disconnect-history l2tp-tunnel upf** command for viewing L2TP tunnel disconnect history.	2026.01.0
Introduced support for L2TP tunnel failover.	2025.03.0
Enhancement Introduced: • Tunnel-Preference attribute • Tunnel-Client-Auth-ID attribute	2025.02.0
First introduced.	2021.04.0

Feature Description

Note This feature is Network Services Orchestrator (NSO) integrated.

Majority of the digital subscriber line (DSL) broadband deployments use PPPoE sessions to provide Subscriber services. These sessions terminate the PPP link and provide all the features, service, and billing on the same node. These sessions are called PPP Terminated (PTA) sessions .

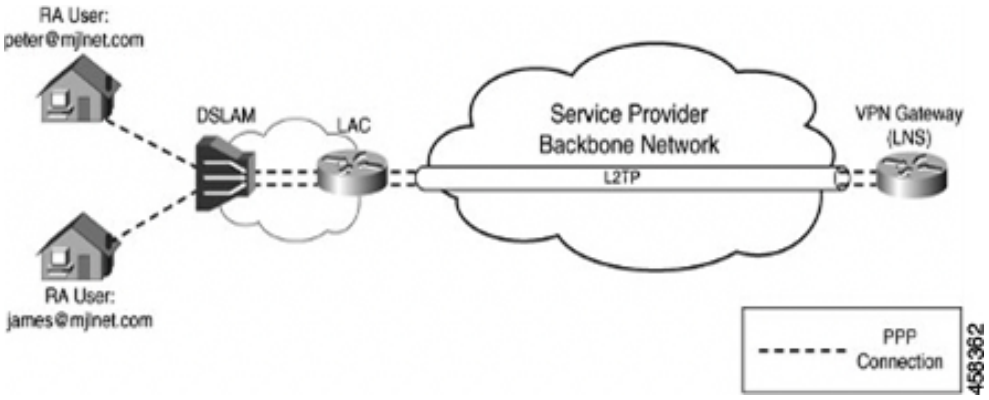
There are some wireline subscriber deployments are in wholesale-retail model where some ISPs work with others to provide the access and core services separately. In such cases, the subscribers are tunneled between wholesale and retail ISPs using the Layer 2 Tunneling Protocol (L2TP) protocol.

L2TP Overview

In cnBNG, the L2TP performs the hand-off task of the subscriber traffic to the Internet service provider (ISP). To do this, L2TP uses two network components:

- L2TP Access Concentrator (LAC)—The L2TP enables subscribers to dial into the L2TP access concentrator (LAC), which extends the PPP session to the LNS. cnBNG provides LAC.
- L2TP Network Server—The L2TP extends PPP sessions over an arbitary netowrk to a remote network server that is, the L2TP network server (LNS). The ISP provides LNS.

The overall network deployment architecture is also known as Virtual Private Dial up Network (VPDN). The overall topology of LAC and LNS is depicted as follows:



The CP for LAC and LNS depend on the L2TP session termination. Developing these two control planes in a single cnBNG microservice has the following benefits:

Simplified Single L2TP Control Plane

- Reduces the configuration complexity of the current XR L2TP

  vpdn-groups, vpdn-templates, l2tp-class and so on are simplified.
- Supports LC subscriber (not supported on the physical BNG)
- Avoids Ns/Nr checkpointing issues of pBNG to support RPFO

Collocated LAC and LNS

- Supports LAC and LNS in the same cnBNG CP, with different User Plane (UPs)
- Enables sharing of the same AAA and Policy Plane
- Simplifies management and troubleshooting

Flexible Deployment Options

The integration of LAC and LNS into a centralized cnBNG CP provides highly flexible deployments options to suit different customer use-cases and needs. For example, the cnBNG CP can host the CP functionality either for a LAC or LNS UP. Also, the same CP cluster can act as a CP for both LAC and LNS UPs from different data centers (or even from the same user-plane, if the user-plane supports it) except for the same session at the same time.

L2TP Features

The cnBNG supports the following Layer 2 Tunneling Protocol (L2TP) features:

- Tunnel authentication
- AVP encryption
- Tunnel Hello interval
- IP ToS value for tunneled traffic
- IPv4 don't fragment bit

- DSL line information attributes
- IPv4 tunnel source address
- IPv4 tunnel destination address
- IPv4 destination load balancing
- Tunnel mode
- MTU for LCP negotiation
- TCP maximum support
- Start-Control-Connection-Request (SCCRQ) timeout
- SCCRQ retries
- Control packet retransmission
- Control packet retransmission retries
- Receive window size for control channel
- Rx and Tx connect speed
- Tunnel VRF
- Tunnel session limit
- Weighed and Equal Loadbalancing
- Tunnel password for authentication
- Domain name and tunnel assignment
- LCP and Authentication renegotiation
- LAC hostnames for tunnelling requests
- Tunnel preference
- Tunnel-Client-Auth-ID support

L2TP tunnel failover support

An L2TP tunnel failover is a network feature that

- maintains subscriber connectivity by switching to a backup LNS server if the preferred LNS server is unavailable
- ensures minimal service interruption during failures, and
- enables seamless broadband service for end users.

Table 3: Feature History

Feature Name	Release Information	Description
L2TP tunnel failover support	2025.03.0	You can now maintain continuous broadband service for subscribers, even if the preferred LNS server becomes unavailable. The L2TP tunnel failover feature automatically switches subscriber sessions to a backup LNS server whenever a failure occurs, minimizing service disruption and ensuring seamless connectivity for end users.

Reliable connectivity with L2TP tunnel failover

The L2TP tunnel failover feature allows the cnBNG CP to provide reliable broadband connectivity. When a preferred L2TP Network Server (LNS) is not available, the system automatically selects the next available server, ensuring continuous service for subscribers.

How L2TP tunnel failover works

This process ensures that remote users always reach their private network, even if a failure occurs at the preferred LNS server.

Summary

The key components involved in the L2TP tunnel failover process are:

- PPPoE client: Initiates a dial-in session to the ISP access server.
- Network Access Server (NAS): Receives PPPoE connection from the client and forwards the session if needed.
- L2TP Access Concentrator (LAC): Establishes and manages L2TP tunnels to available LNS servers.
- L2TP Network Server (LNS): Authenticates and connects the subscriber to the private network.

Workflow

The process involves these stages:

1. The PPPoE client connects to the NAS, which determines if the session should be forwarded to an LNS via L2TP.
2. The LAC establishes L2TP tunnels to one or more LNS servers, maintaining a list of available servers.
3. When the preferred LNS server is unavailable (due to failure or lack of response), the system marks it as DOWN and starts a timer.
4. The LAC selects the next available LNS server from its list, using either Tunnel-Preference or Load-Balancing algorithms, to re-establish the subscriber session.
5. If the original LNS server becomes available again, it is marked as ACTIVE and may be used for new sessions.

Result

Subscribers experience uninterrupted broadband services as the system automatically redirects their sessions to available LNS servers during failures.

Configure tunnel busy timeout for LNS servers

Configure the amount of time that a failed LNS server remains marked as DOWN before being eligible for reconnection attempts.

Before you begin

- Ensure you have access privileges to configure cnBNG Control Plane.
- Know the profile name you wish to configure.

Follow these steps to configure the tunnel busy timeout:

Procedure

Step 1 Set the tunnel busy timeout value in seconds.

Example:

```
profile l2tp profile-name
  tunnel timeout busy value
exit
```

The system waits for the configured period before attempting to reuse an LNS server that was previously marked as DOWN.

This is a sample configuration.

```
profile l2tp prof1
  tunnel timeout busy 77
exit
```

Note We recommend setting the tunnel busy timeout to a value that balances quick failover with the possibility of LNS server recovery. A shorter timeout enables faster recovery if the LNS server returns quickly, while a longer timeout avoids repeated failed attempts during outages.

NOTES:

- **profile l2tp** *profile-name*: Specifies the PPPoE profile name and enters the Profile L2TP mode.
- **tunnel timeout busy** *value*: Specifies the timeout period before marking an LNS server as busy. The valid values range from 60 to 65535 seconds. Default value: 60 seconds.

Step 2 Use the **show l2tp-tunnel-destination upf** *upf_name* command to view the status (active or down) of LNS servers.

Example:

```
bng# show l2tp-tunnel-destination upf lps_asr9k-200
Wed Aug  13 10:57:06.504 UTC+00:00
result
[
  {
    "server": "192.69.1.199",
    "vrf": "",
    "status": "Active"
  }
]
```

Seamless L2TP session continuity during server reboot

Seamless L2TP session continuity during server reboot is a high-availability feature that

- allows ISPs to provide uninterrupted broadband services to subscribers during an L2TP server reboot
- automatically enables the secondary L2TP instance to manage existing sessions for affected User-Plane Functions (UPFs) until the primary instance is available, and
- restores session management to the primary instance once it returns to service.

Table 4: Feature history table

Feature name	Release information	Feature description
Seamless L2TP session continuity during server reboot	2026.01.0	You can now maintain uninterrupted broadband services for subscribers during an L2TP server reboot. If the primary server restarts, a secondary L2TP instance automatically manages active sessions for affected UPFs until the primary returns, ensuring session continuity and consistent service.

Supported platforms and deployment scenarios

This feature is supported in these cnBNG Control Plane deployments:

- All-in-One (AIO) server deployments (with or without geographical redundancy (GR) and subscriber redundancy group (SRG))
- Three-server and four-server deployments (with or without GR and SRG)
- Deployments using OpenShift

L2TP Pod instance limitations

- Only a single L2TP pod instance restart is supported within a specified time window.
- Double fault scenarios, where one L2TP instance is down due to a server reboot and the other L2TP instance restarts during that period, are not supported.
- Split-brain scenarios, where inter-pod connectivity is lost between L2TP pods but both pods remain reachable from the rest of the system, are not supported.

IETF Tagged Attributes on LAC

The IETF Tagged Attributes support on L2TP Access Concentrator (LAC) provides a means of grouping tunnel attributes referring to the same tunnel in an Access-Accept packet sent from the RADIUS server to the LAC. The Access-Accept packet can contain multiple instances of same RADIUS attributes, but with different tags. The tagged attributes support ensures that all attributes pertaining to a given tunnel contain the same value in their respective tag fields, and that each set includes an appropriately-valued instance of the Tunnel-Preference attribute. This conforms to the tunnel attributes that are to be used in a multi-vendor network environment, thereby eliminating interoperability issues among Network Access Servers (NASs) manufactured by different vendors.

For details of RADIUS Attributes for Tunnel Protocol Support, refer RFC 2868.

These examples describe the format of IETF Tagged Attributes:

```
Tunnel-Type = :0:L2TP, Tunnel-Medium-Type = :0:IP, Tunnel-Server-Endpoint = :0:"1.1.1.1",
Tunnel-Assignment-Id = :0:"1", Tunnel-Preference = :0:1, Tunnel-Password = :0:"hello"
```

A tag value of 0 is used in the above example in the format of `:0:`, to group those attributes in the same packet that refer to the same tunnel. Similar examples are:

```
Tunnel-Type = :1:L2TP, Tunnel-Medium-Type = :1:IP, Tunnel-Server-Endpoint = :1:"2.2.2.2",
Tunnel-Assignment-Id = :1:"1", Tunnel-Preference = :1:1, Tunnel-Password = :1:"hello"

Tunnel-Type = :2:L2TP, Tunnel-Medium-Type = :2:IP, Tunnel-Server-Endpoint = :2:"3.3.3.3",
Tunnel-Assignment-Id = :2:"1", Tunnel-Preference = :2:2, Tunnel-Password = :2:"hello"

Tunnel-Type = :3:L2TP, Tunnel-Medium-Type = :3:IP, Tunnel-Server-Endpoint = :3:"4.4.4.4",
Tunnel-Assignment-Id = :3:"1", Tunnel-Preference = :3:2, Tunnel-Password = :3:"hello"

Tunnel-Type = :4:L2TP, Tunnel-Medium-Type = :4:IP, Tunnel-Server-Endpoint = :4:"5.5.5.5",
Tunnel-Assignment-Id = :4:"1", Tunnel-Preference = :4:3, Tunnel-Password = :4:"hello"

Tunnel-Type = :5:L2TP, Tunnel-Medium-Type = :5:IP, Tunnel-Server-Endpoint = :5:"6.6.6.6",
Tunnel-Assignment-Id = :5:"1", Tunnel-Preference = :5:3, Tunnel-Password = :5:"hello"
```

Table 5: Supported IETF Tagged Attributes

IETF Tagged Attribute Name	Value	Type
Tunnel-Type	integer	64
Tunnel-Medium-Type	integer	65
Tunnel-Client-Endpoint	string	66
Tunnel-Server-Endpoint	string	67
Tunnel-Password	string	69
Tunnel-Assignment-ID	string	82
Tunnel-Preference	integer	83
Tunnel-Client-Auth-ID	string	90
Tunnel-Server-Auth-ID	string	91

Tunnel-Preference support

Tunnel-Preference is a RADIUS attribute that

- enables RADIUS servers to communicate tunnel preference selection information
- allows operators to prioritize or prefer certain tunnels over others when multiple tunnels are available for data transmission.

Table 6: Feature History

Feature Name	Release Information	Description
Tunnel-Preference support	2025.02.0	You can now prioritize specific tunnels for data transmission when multiple tunnels are available. The Tunnel-Preference attribute in RADIUS messages allows you to specify the preferred tunnel, offering greater flexibility in tunnel selection.

Tunnel-Preference attribute overview

Tunnel preference is communicated via the Tunnel-Preference attribute in RADIUS messages. This attribute indicates the preference that is given to a specific tunnel. The tunnel with the lowest numerical value in the **Value** field receives the highest preference, with 0x000000 being the most preferred and 0xFFFFFF being the least preferred.

Handling multiple tunneling attributes

If multiple sets of tunneling attributes are returned by the RADIUS server, the Tunnel-Preference attribute should be included in each set to indicate the relative preference assigned to each tunnel.

Attribute tagging and grouping

The Tunnel-Preference attribute includes a Tag field that groups attributes referring to the same tunnel.

Tunnel selection logic in L2TP with Tunnel-Preference and load balancing

In the L2TP tunnel pod, if a tunnel-preference value is received from the RADIUS server, it overrides any existing load-balancing configuration.

If multiple tunnels have the same tunnel-preference value, the system checks whether a load-balancing method, such as **weighted** or **equal**, is configured:

- If a load-balancing method is configured, the tunnel is selected based on the load-balancing method
- If no load-balancing method is configured, the system randomly selects a tunnel from those with the same preference value.

Tunnel-Preference attribute fields

Attribute name	Type	Length	Tag	Value
Tunnel-Preference	83	6	One octet in length. Valid values range from 0x01 to 0x1F. If the Tag field is not used, it must be set to zero (0x00).	Three octets in length. Higher preference is given to lower values, with 0x000000 being most preferred and 0xFFFFFF least preferred.

Benefits of Tunnel-Preference attribute

- Allows service providers to prioritize specific tunnels.
- Overrides load balancing configurations to ensure traffic flows through the preferred tunnel.
- Provides flexibility in tunnel selection based on RADIUS server directives.

Configure Tunnel-Preference attribute

You can configure the **Tunnel-Preference** attribute using the user-profile on the RADIUS server. You can then directly receive it from the RADIUS server. No configuration is needed on the cnBNG CP side.

Procedure

Define the **Tunnel-Preference** attribute on the RADIUS server.

Example:

```
RADIUS:
user@example.com    Password="abc"
        Service-Type = Outbound-User,
        Tunnel-Type = :1:L2TP,
        Tunnel-Medium-Type = :1:IP,
        Tunnel-Client-Endpoint = ":1:10.0.0.201",
        Tunnel-Server-Endpoint = ":1:10.1.10.129",
        Tunnel-Client-Auth-Id = ":1:LAC1",
        Tunnel-Server-Auth-Id = ":1:LNS1",
        Tunnel-Assignment-Id = ":1:one",
        Tunnel-Password = ":1:cisco1",
        Tunnel-Preference = :1:100
        Tunnel-Type = :2:L2TP,
        Tunnel-Medium-Type = :2:IP,
        Tunnel-Client-Endpoint = ":2:10.0.0.201",
        Tunnel-Server-Endpoint = ":2:10.1.10.130",
        Tunnel-Client-Auth-Id = ":2:LAC2",
        Tunnel-Server-Auth-Id = ":2:LNS2",
        Tunnel-Assignment-Id = ":2:two",
```

```
Tunnel-Password = ":2:cisco2",
Tunnel-Preference = :2:200
```

Tunnel-Client-Auth-Id support

Tunnel-Client-Auth-ID is a RADIUS attribute that

- specifies the name used by the tunnel initiator during the authentication phase of tunnel establishment
- is defined in RFC 2868 - *RADIUS Attributes for Tunnel Protocol Support.*

Table 7: Feature History

Feature Name	Release Information	Description
Tunnel-Client-Auth-ID support	2025.02.0	This attribute supports tunneling protocols by specifying the authentication name used by the tunnel inititator during the authentication phase of tunnel establishment.

Tunnel-Client-Auth-ID attribute fields

Attribute	Type	Length	Tag	String
Tunnel-Client-Auth-ID	90 - The identifier for Tunnel-Client-Auth-ID.	Must be greater than or equal to 3 bytes.	One octet in length; groups attributes for the same tunnel: Values between 0x01 and 0x1F indicate the specific tunnel, while values above 0x1F are interpreted as the first byte of the String field.	Mandatory field containing the authentication name of the tunnel initiator, represented in UTF-8 charset.

Example of Tunnel-Client-Auth-ID attribute in a RADIUS message

In a typical **Access-Request** or **Access-Accept** message, the Tunnel-Client-Auth-Id is included in the attributes field, for example,

- `Tunnel-Client-Auth-Id: " :1:CSCO_LAC"`

  Here, " CSCO_LAC" is the unique identifier for a client inside the tunnel.

Configure Tunnel-Client-Auth-ID attribute

You can configure the **Tunnel-Client-Auth-Id** attribute using the user-profile on the RADIUS server. You can then directly receive it from the RADIUS server. No configuration is needed on the cnBNG CP side.

Procedure

Step 1 Define the **Tunnel-Client-Auth-Id** attribute on the RADIUS server.

Example:

```
RADIUS:
user@example.com    Password="abc"
        Service-Type = Outbound-User,
        Tunnel-Type = :1:L2TP,
        Tunnel-Medium-Type = :1:IP,
        Tunnel-Client-Endpoint = ":1:10.0.0.201",
        Tunnel-Server-Endpoint = ":1:10.1.10.129",
        Tunnel-Client-Auth-Id = ":1:LAC1",
        Tunnel-Server-Auth-Id = ":1:LNS1",
        Tunnel-Assignment-Id = ":1:one",
        Tunnel-Password = ":1:cisco1",
        Tunnel-Preference = :1:100
        Tunnel-Type = :2:L2TP,
        Tunnel-Medium-Type = :2:IP,
        Tunnel-Client-Endpoint = ":2:10.0.0.201",
        Tunnel-Server-Endpoint = ":2:10.1.10.130",
        Tunnel-Client-Auth-Id = ":2:LAC2",
        Tunnel-Server-Auth-Id = ":2:LNS2",
        Tunnel-Assignment-Id = ":2:two",
        Tunnel-Password = ":2:cisco2",
        Tunnel-Preference = :2:200
```

Step 2 Use the **show l2tp-tunnel detail** command to view the configured attributes.

Example:

```
bng# show l2tp-tunnel detail
tunnel-details
{
  "tunResponses": [
    {
      "state": "established",
      "profileName": "l2tp-prof1",
      "tunnelType": "lac",
      "sessionCount": 10,
      "IDs Allocated": 10,
      "routerID": "asr9k-1",
      "srcIP": "10.0.0.201",
      "dstIP": "10.1.10.129",
      "tunnelAssignmentID": "one",
      "localTunnelID": 49078,
      "remoteTunnelID": 51810,
      "tunnelClientAuthID": "LAC1",
      "tunnelServerAuthID": "bng-lns"
    }
  ]
}
```

How it Works

This section provides a brief of how the L2TP Subscriber Management feature works.

L2TP Handling

Both LAC and LNS sessions use L2TP protocol for negotiation and creation of L2TP sessions. However for LAC sessions, there is additional PPPoE handling. This section focuses on the L2TP protocol handling.

LAC Sessions

For LAC sessions, the PPP sessions are terminated on a different network node from where the PPPoE sessions are terminated. The PPPoE sessions are terminated on the LAC, but the PPP session is terminated on an LNS upstream, over an L2TP tunnel. Initial PPP negotiations are done on the LAC to determine the appropriate LNS to tunnel the session. When the tunnel has been established, all PPP handling is handed off to the LNS.

- The PPPoE protocol is negotiated in the same way as a PTA session.
- PPPoE service handles all PPPoE packets and the nitial LCP and authorization packets.
- After authentication , if the user-profile contains service=outbound, PPPoE service decides to tunnel the sessions.
- It reaches out the L2TP pod to initiate a L2TP tunnel . The L2TP tunnel pod creates the tunnel and returns the L2TP session ID.
- The PPPoE service continues to handle the L2TP session FSM and bring-up the LAC session and program the UP via the Subscriber Manager.

LNS Sessions

LNS sessions are similar to PTA sessions in the overall functionality. Instead of PPPoE protocol, here the First-Sign-Of-Life (FSOL) packets are the L2TP Incoming-Call-Request (ICRQ) messages. When the L2TP session protocol is up, then the existing PPP protocol finite state machines (FSM) is triggered to bring up and program the session on the UP.

- L2TP Tunnel pod receives tunnel-create request from the remote LAC.
- After Tunnel is up, PPPoE Pod receives ICRQ to create a session.
- PPPoE pod communicates with the L2TP to get L2TP session-id for the given tunnel ID.
- L2TP generates the session ID and checks the session count.
- PPPoE pod checks if there is forced renegotiation configured for the session. Else, it proceeds with the session programming to the UP.

AAA Attributes for L2TP

The following is the list of AAA attributes for L2TP LAC and LNS sessions.

IETF Attribute: AAA_TUNNEL_PASSWORD (69)

Tunnel-Password=<16byte-encrypted-value>

The value of this attribute is defined as an "encrypted-string". RADIUS decrypts the value and sends a plain-text password to the Subscriber Manager (SM).

For more L2TP IETF Attributes, see IETF Tagged Attributes on LAC, on page 8.

CISCO-VSA: AAA_AT_L2TP_TUNNEL_PASSWORD

Cisco-AVPair += "l2tp-tunnel-password=<plain-text>"

The value of this attribute is defined in "plain-text". RADIUS passes the value to SM in the respective Access-accept request.

If required, the RADIUS server can this as an "encrypted-cisco-visa(36)", which is similar to the Layer1 vendor-specific attributes (VSAs).

In that case, RADIUS-Ep decrypts the complete VSA and sends the plain-text value.

For more L2TP VSA attributes, see RADIUS Vendor-Specific Attributes.

Handling L2TP Sessions during CP-GR Switchover

This feature enables you to manage Layer 2 Tunneling Protocol (L2TP) sessions and tunnels effectively during a CP-GR switchover event. This ensures seamless transition and minimal disruption in connectivity for users.

Table 8: Feature History

Feature Name	Release Information	Description
Handling L2TP Sessions during CP-GR Switchover	2025.02.0	This feature minimizes downtime and ensures smooth transitions during network changes by disconnecting existing L2TP sessions and tunnels during a CP-GR switchover. It then establishes new sessions on new tunnels.

Here's how L2TP sessions are handled during CP-GR switchover:

1. **Disconnection of existing sessions and tunnels:**

   When a CP-GR switchover gets triggered for a specific instance-id, all L2TP sessions and the corresponding L2TP tunnels on UPFs associated with that instance-id are disconnected.

2. **Establishment of new sessions and tunnels:**

   Following the CP-GR switchover, new L2TP sessions are established on new tunnels.

Call Flows

This section includes the following high-level call flows.

LAC Session Bringup Call Flow

The LAC Session Bringup call flow is as follows.

Figure 1: LAC Session Bringup Call Flow

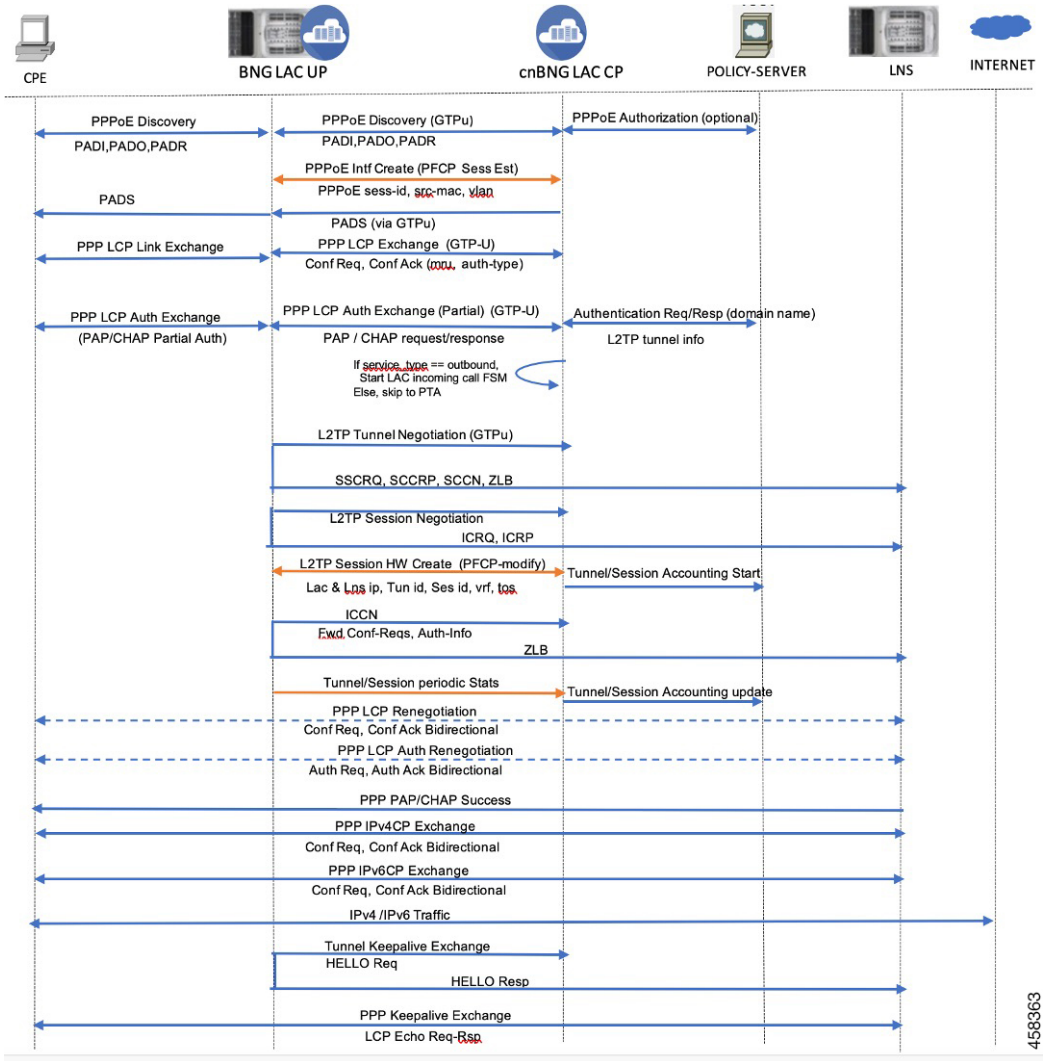


Table 9: LAC Session Bringup Call Flow Description

Steps	Description
1	On learning the first control packet, the BNG-CP sends a Session Creation request to create a new packet forwarding state for the data packet. This updates the BNG-UP state. **Note** At this step, it is possible to create a session from the redirected control packet. By doing so, resources are consumed on the BNG-UP to allow individual subscriber control packet management such as blocking, rate limiting, and specific packet filtering. It is also possible to postpone the session creation. By doing so, additional resources BNG-UP are not consumed, but individual subscriber control packet management is not possible.

Steps	Description
2	The BNG-UP sends the following response to the BNG-CP: • Informs that the states are installed. • Informs that it (BNG-UP) is ready to forward the subscriber PPP control packets.
3	The BNG-CP sends the PADO message back to the CPE through the BNG-UP using the control packet redirect interface.
4	The PADR message is sent from the CPE through the BNG-UP using the control packet redirect interface.
5	The BNG-CP sends the PADS message back to the CPE through the BNG-UP using the control packet redirect interface.
6	The LCP configuration request is sent from the CPE through the BNG-UP using the control packet redirect interface.
7	The BNG-CP sends the LCP configuration acknowledgement back to the CPE through the BNG-UP using the control packet redirect interface. The LCP configuration acknowledgement indicates either a PAP or CHAP authentication challenge. Options: • Option 1: If the client chooses PAP, the CPE sends a PAP request to the BNG-CP through the BNG-UP using the control packet redirect interface. The PAP password is sent as an Access request to the AAA server. • Option 2: If CHAP is required, the BNG-CP initiates a challenge to the CPE though the BNG-UP using the control packet redirect interface. The CPE responds back to the challenge to the BNG-CP. The challenge is sent to the AAA server.
8	The AAA successfully authenticates the CPE and replies to the CPE with a PAP/CHAP success and that this is a L2TP session. **Note** If the RADIUS profile received in AAA Accept-Ack has the field “service-type” with the value as “outbound-user”, this means that the session must be tunneled to the LNS IP address (either specified in the same profile or available in the Control Plane configuration).
9	The BNG-CP sends a Session Establishment message to the BNG-UP. The BNG-CP programs the NBG-UP control packet redirect rules to do the following: • Decapsulate and send the L2TP control message towards the LNS. • Redirect L2TP control message back to the BNG-CP. This session establishment is only on a per-tunnel basis.
10	The BNG-UP sends the following response to the BNG-CP: • Informs that the states are installed. • Informs that it (BNG-UP) is ready to forward the L2TP control packets.

Steps	Description
11	The BNG-CP sends Start-Control-Connection-Request (SCCRQ), Start-Control-Connection-Reply (SCCRP), Start-Control-Connection-Connected (SCCCN), and Zero-Length Body (ZLB) to the LNS via the BNG-UP through the control packet redirect interface.
12	The BNG-CP sends Incoming-Call-Request (ICRQ), Incoming-Call-Reply (ICRP), Incoming-Call-Connected (ICCN), and ZLB to the LNS via the BNG-UP through the control packet redirect interface.
13	The BNG-CP sends a Session Modify request if there is a previous session established to allow for data packet forwarding to the LNS (and control packet if not done already). If a previous session was not established, this is a Session Request message to allow for data packet forwarding to the LNS. This updates the User Plane state. **Note** Subscriber session creation can be performed at any steps prior to this. This step is the last chance for a session creation to avoid subscriber data packets drops. Immediately after this step, the CPE is assigned an address and data packets would be sent immediately.
14	The BNG-UP sends the following response to the BNG-CP: • Informs that the states are installed. • Informs that it (BNG-UP) is ready to forward subscribers PPP control and data packets.
15	If the LNS has cached the LCP configuration and there is no negotiation disagreement, this step can be skipped. If LNS has not cached the LCP configuration or the session requires renegotiation, then the LCP negotiation takes place.
16	If the LNS has cached the authentication information and there is no disagreement on authentication, this step can be skipped. If LCP has not cached the authentication information or authentication has failed, then reauthorization occurs.
17	The IP Control Protocol (IPCP) takes place between the CPE and the LNS through the BNG-UP.
18	The PPP LCP echo hello are exchanged between the CPE and the LNS through the BNG-UP.

LAC Session Bringdown Call Flow

The LAC Session Bringdown call flow is as follows.

Figure 2: LAC Session Bringdown Call Flow

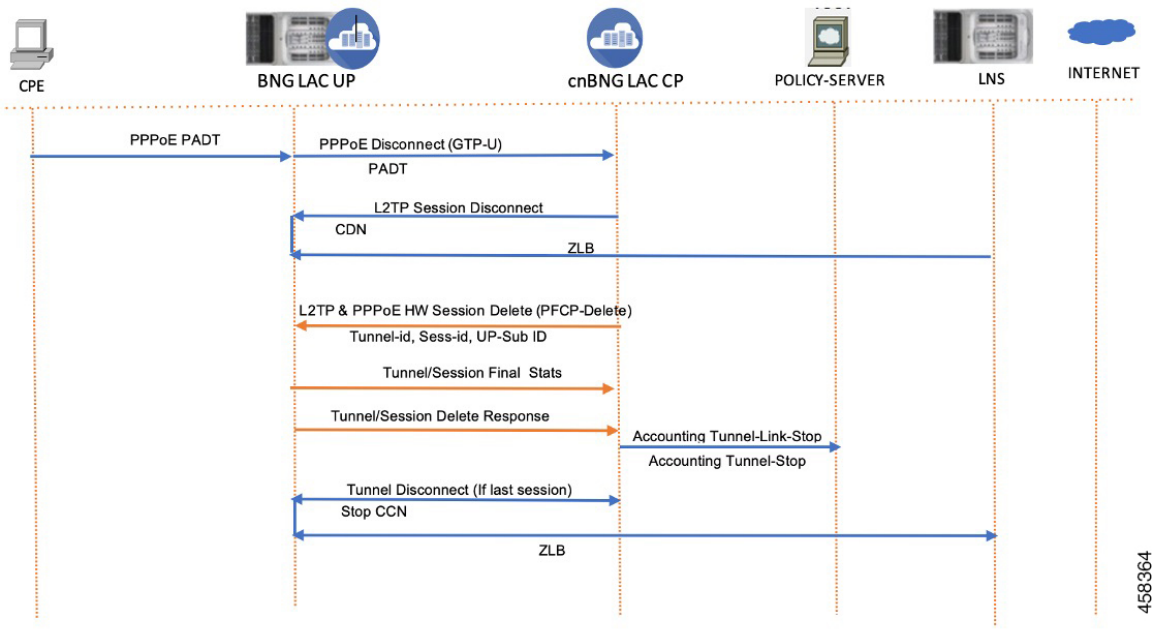


Table 10: LAC Session Bringdown Call Flow Description

Steps	Description
1	The L2TP (LAC) session and tunnel bringdown can occur due to various reasons. For example, CPE can send PADT to gracefully bringdown the subscriber session. This triggers the L2TP session cleanup between LAC and LNS.
2	If it is the last session in the L2TP tunnel, the tunnel is also deleted and the PPPoE session is cleaned up in the LAC. The session or tunnel bringdown occurs in the following scenarios: • PPP keepalive failure between CPE and LNS. • Tunnel keepalive failure. In this case, all sessions in the tunnel are removed first. • Admin clear on either LAC or LNS.

LNS Session Bringup Call Flow

The LNS Session Bringup call flow is as follows.

Figure 3: LNS Session Bringup Call Flow

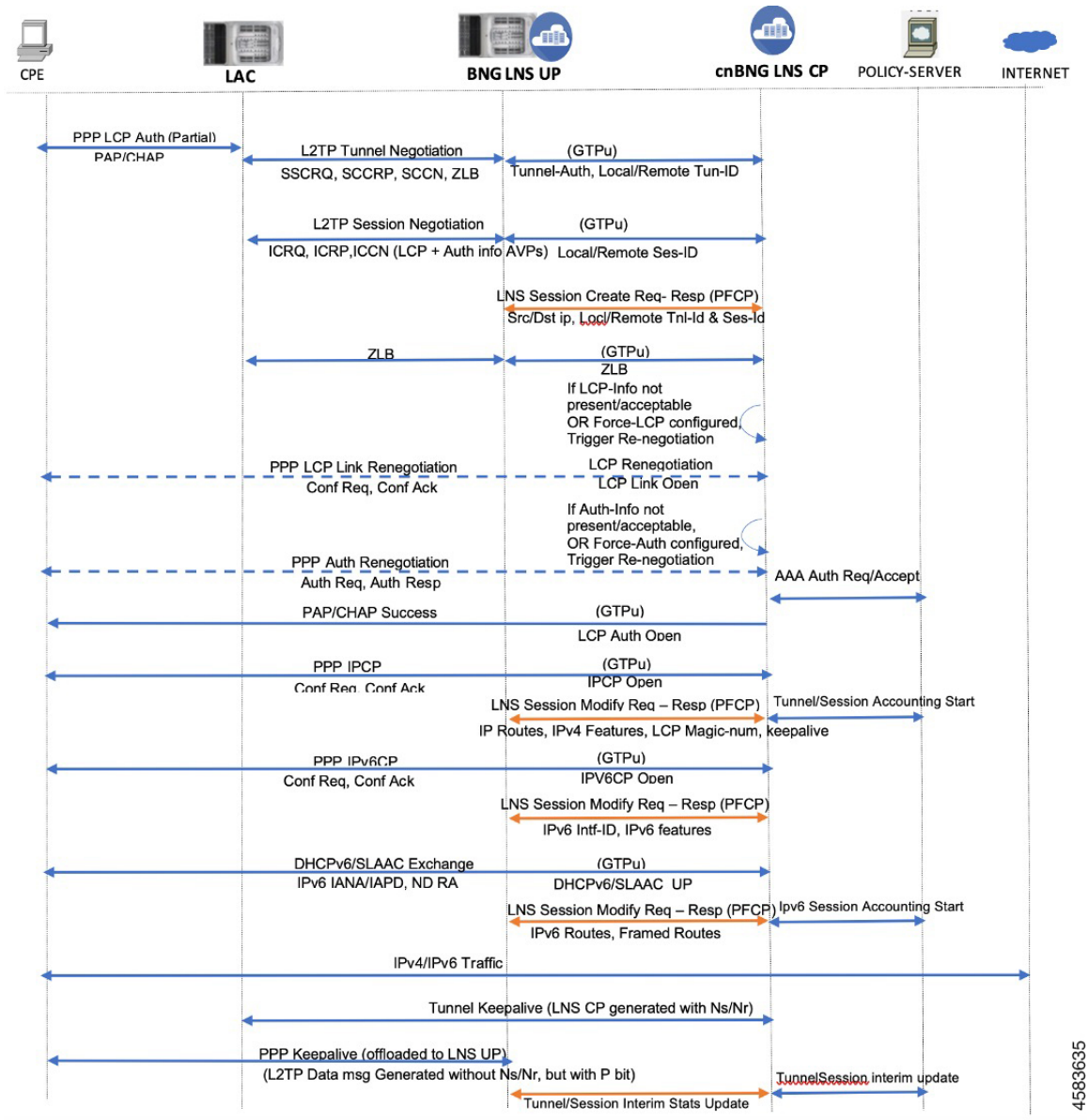


Table 11: LNS Session Bringup Call Flow Description

Steps	Description
1	The Start Control Connection Request (SCCRQ) message is received through the control packet redirect interface following the common packet redirect rule.
2	The BNG-CP sends a Session Establishment request message to the BNG-UP. The BNG-CP programs the DBNG-UP control packet redirect rules to send L2TP control message towards the BNG-CP to only accept particular tunnels.

Steps	Description
3	The BNG-UP sends the following response back to the BNG-CP: • Informs that the states are installed. • Informs that it (BNG-UP) is ready to forward the L2TP control packets.
4	The BNG-CP exchanges Start Control Connection Reply (SCCRP), Start Control Connection Connected (SCCCN), and Zero Length Body (ZLB) with the LAC using the control packet redirect interface.
5	The BNG-CP receives the Incoming Call Request (ICRQ) message (includes AVP defined in RFC 5515).
6	After receiving the ICRQ message, the BNG-CP has the L2TP session ID information. The BNG-CP can send a Session Establishment request to the BNG-UP to ensure only known L2TP sessions are accepted. 1 **Note** At this step, it is possible to create a session from the redirected control packet. By doing so, resources are consumed on the DBNG-UP in order to allow individual subscriber control packet management such as blocking, rate limiting, and specific packet filtering. It is also possible to postpone the session creation. By doing so, additional resources DBNG-UP are not consumed, but individual subscriber control packet management is not possible
7	The BNG-UP sends the following response back to the BNG-CP: • Informs that the states are installed. • Informs that it (BNG-UP) only accepts L2TP control packet from known sessions.
8	The BNG-CP exchanges ICRP, ICCN, and ZLB with the LAC using the control packet redirect interface.
9	If the LNS has cached the LCP configuration and there is no negotiation disagreement, this step can be skipped. If the LCP has not cached the LCP configuration or the session requires renegotiation, then the LCP negotiation takes place.
10	If the LNS has cached the authentication information and there is no disagreement on authentication, this step can be skipped. If LCP has not cached the authentication information or authentication has failed, then reauthorization occurs.

Steps	Description
11	After authentication, the BNG-CP knows the IP address or prefix (or both) for the subscriber either though the local address server or from the AAA returned VSAs. The BNG-CP sends a Session Modify request if there is already an established session to update the User Plane (UP) state. If there are no prior sessions, this requires a Session Establishment request to update the UP. **Note** Subscriber session creation can be performed at any steps prior to this. This step is the last chance for a session creation to avoid subscriber data packets drops. Immediately after this step, the CPE is assigned an address and data packets would be sent immediately.
12	The BNG-UP sends the following response back to the BNG-CP: • Informs that the states are installed. • Informs that it (BNG-UP) is ready to forward subscribers PPP control and data packets.
13	The IPCP takes place between the CPE and the LNS through the BNG-UP.
14	The PPP LCP echo hello are exchanged between the CPE and the LNS through the BNG-UP.

[1] At this step, it is possible to create a session from the redirected control packet. By doing so, resources are consumed on the BNG-UP to allow individual subscriber control packet management such as blocking, rate limiting, and specific packet filtering. It is also possible to postpone the session creation. By doing so, additional resources BNG-UP are not consumed, but individual subscriber control packet management is not possible.

LNS Session Bringdown Call Flow

The LNS Session Bringdown call flow is as follows.

Figure 4: LNS Session Bringdown Call Flow

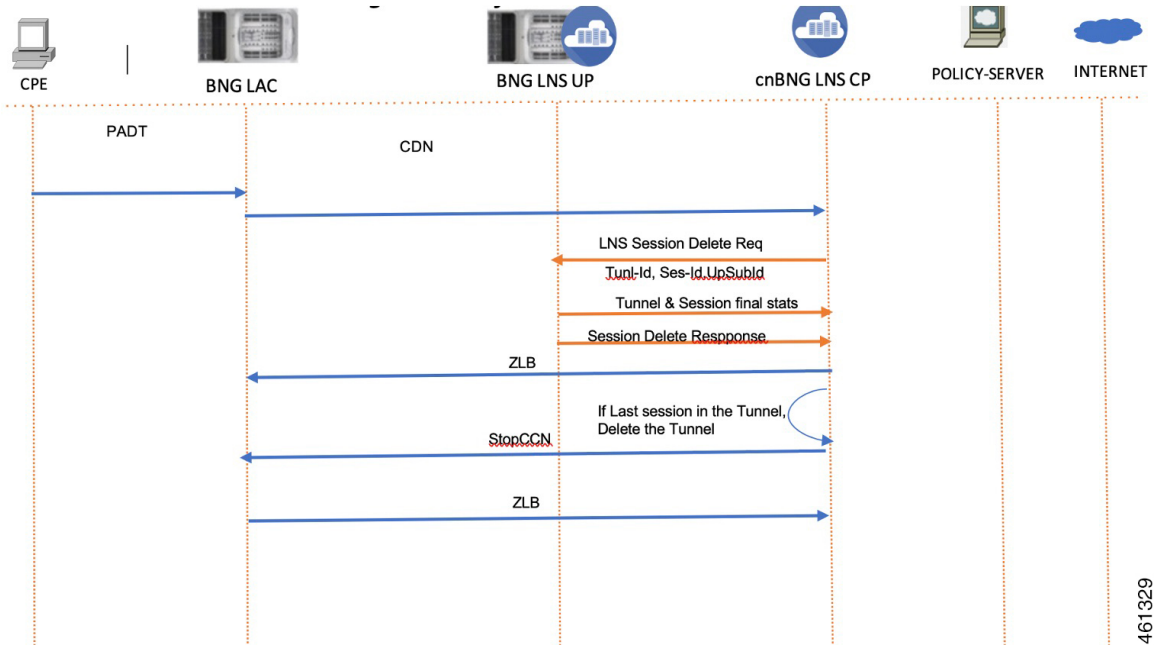


Table 12: LNS Session Bringdown Call Flow Description

Steps	Description
1	LAC sends a Call-Disconnect-Notify (CDN) message to release the session on the LNS.
2	cnBNG CP deletes the session on UP. It releases all the resources and collects the final statistics from the UP and sends the Accounting-Stop message.
	cnBNG CP sends ZLB as ackndoledgement.
	If it is the last sesion on the tunnel, cnBNG CP sends a Stop-Control-Connection-Notification (Stop-CCN) message to bring down the tunnel.

Standard Compliance

The L2TP Subscriber Management feature is aligned with the following standard:

- RFC 2661: Layer Two Tunneling Protocol "L2TP"

Limitations

The LT2P Subscriber Management feature has the following limitations:

- On-the-fly changes to L2TP profile is not supported.
- L2TP attributes should be configured only for session-activate event.

- Tunnel load balancing with Tunnel-Assignment-ID is not supported.
- Weighted Tunnel load balancing can be configured only in the profile.
- The TCP maximum segment size (TCP-MSS) is supported at the global User Plane Function (UPF) chassis level and not at the tunnel or session level. It must be configured on the ASR 9000 UPF.

Configuring the L2TP Subscriber Management Feature

This section describes how to configure the L2TP Subscriber Management feature.

Configuring the L2TP Subscriber Management feature involves the following step:

Creating the L2TP profile

Creating the L2TP Profile

Use the following commands to create the Layer2 Tunnelling Protocol (L2TP) profile and provide the L2TP specific parameters.

```
config
  profile l2tp l2tp_profile_name
    authentication

    encrypt-avp
    hello-interval interval_in_seconds
    hostname local_hostname
    ip-tos { ip_tos_value | reflect }
    ipv4 { df-bit { reflect | set } | source ip_address }
    mode lac
      domain domain_name [ tun-assign-id tunnel_id ]
      dsl-info-forwarding
      ipv4 { destination ip_address | df-bit { reflect | set } |
          source ip_address }
      rx-connect-speed kbps
      tunnel-load-balancing { equal | weighted }
      tx-connect-speed kbps
    mode lns
      force-lcp-renegotiation
      mtu mtu_value
      terminate-from remote_hostname
    password password
    receive-window number_of_packets
    retransmit { retries number_of_retries |
    timeout { max max_timeout | min min_timeout }
    tcp adjust-mss mss_value
    tunnel { session-limit number_of_sessions |
    timeout { no-session timeout_value | busy value }
    vrf vrf_name
    exit
```

NOTES:

- **profile l2tp** *l2tp_profile_name*: Specifies the PPPoE profile name and enters the Profile L2TP mode.
- **authentication**: Enables L2TP tunnel authentication.
- **congestion-control**:Enables L2TP congestion control.
- **encrypt-avp**: Hides attribute-value pair (AVPs) in outgoing control messages.
- **hello-interval** *interval_in_seconds*: Sets the hello interval in seconds. The valid values range from 10 to 1000 seconds.
- **hostname** *local_hostname*: Specifies the local hostname of the tunnel. The valid value is an alphanumeric string ranging from 1 to 256. The name of the Control Plane (CP) is the default local hostname.
- **ip-tos {** *ip_tos_value* **| reflect }**: Sets the IP Type of Service (ToS) value for tunneled traffic. The ToS valid values range from 1 to 255. The control packets use 0xC0 as the default value.
- **ipv4 { destination** *ip_address* **| df-bit { reflect | set } | source** *ip_address* **}**: Specifies the IPv4 settings for the tunnel:
  - **df-bit { reflect | set }**: Specifies the IPv4 Don't Fragment (DF) bit.

    reflect: Reflects the DF bit from the specified inner IP address.

    set: Sets the DF bit.
  - **source** *ip_address*: Specifies the source IP address of the tunnel.
- **mode { lac | lns }**: Configures LAC or LNS.
  - **mode lac { domain** *domain_name* **[tun-assign-id** *tunnel_id* **] | dsl-info-forwarding | ipv4 { destination** *ip_address* **| df-bit { reflect | set } | source** *ip_address* **} | rx-connect-speed** *kbps* **| tunnel-load-balancing { equal | weighted } | tx-connect-speed** *kbps*: Configures a L2TP Access Concentrator (LAC) to request the establishment of an L2TP tunnel to an L2TP Network Server (LNS).
    - **domain** *domain_name* **[tun-assign-id** *tunnel_id* **]**: Specifies the domain name to match. The valid values range from 1 to 255. The control packets use 0xC0 as the default value.

      tun-assign-id *tunnel_id*: Specifies the domain name with a tunnel ID.
    - **dsl-info-forwarding**: Forwards DSL line information attributes.
    - **ipv4 { destination** *ip_address* **| df-bit { reflect | set } | source** *ip_address* **}**: Specifies the IPv4 settings for the tunnel:
      - **destination** *ip_address*: Specifies the destination IP address of the tunnel.
      - **df-bit { reflect | set }**: Specifies the IPv4 Don't Fragment (DF) bit.

        reflect: Reflects the DF bit from the specified inner IP address.

        set: Sets the DF bit.
      - **source** *ip_address*: Specifies the source IP address of the tunnel.
    - **rx-connect-speed** *kbps*: Specifies the receiving (Rx) connection speed in kbps. The valid values range from 9 to 100000000 kbps.

- **tunnel-load-balancing { equal | weighted }** : Specifies equal or weighted load sharing of the tunnel.
- **tx-connect-speed** *kbps*: Specifies the transmitting (Tx) connection speed in kbps. The valid values range from 9 to 100000000 kbps.

- **mode lns { force-lcp-renegotiation | mtu | terminate-from** *remote_hostname*: Configures a LNS to accept requests from LAC to establish L2TP tunnel:
  - **force-lcp-renegotiation**: Forces Link Control Protocol (LCP) and Authorisation renegotiation.

    mtu *mtu_value*: Specifies the MTU for LCP negotiation. The *mtu_value* valid values range from 500 to 2000. The default value is 1492.
  - **terminate-from** *remote_hostname*: Specifies the hostname of the remote peer to accept tunnels.

- **password** *password*: Specifies the password for tunnel authentication.
- **receive-window** *number_of_packets*: Specifies the receive window size for the tunnel. The valid values range from 1 to 5000 packets. The default value is 4.
- **retransmit { retries** *number_of_retries* **| timeout { max** *max_timeout* **| min** *min_timeout* **}**: Specifies the control message retransmission parameters.
  - **retries** *number_of_retries*: Specifies the maximum number of retries for control packets.
  - **timeout { max** *max_timeout* **| min** *min_timeout* **}**: Specifies the control packet retransmission timeout parameters.

    max *max_timeout*: Specifies the control packet retransmission maximum timeout parameters. The valid values range from 1 to 8 seconds. The default value is 8.

    min *min_timeout*: Specifies the control packet retransmission minimum timeout parameters. The valid values range from 1 to 8 seconds. The default value is 1.

- **tcp adjust-mss** *mss_value*: Adjusts the TCP Maximum Segment Size (MSS) value of TCP SYN (synchronize) packets. The valid values range from 500 to 1500 packets.
- **tunnel { session-limit** *number_of_sessions* **| timeout { no-session** *timeout_value* **| busy** *value* **}**: Limits the sessions for a tunnel or deletes the tunnel after timeout
  - **session-limit** *number_of_sessions*: Specifies the maximum number of L2TP sessions per tunnel. The valid values range from 1 to 64000 sessions.
  - **timeout { no-session** *timeout_value* **}**: Specifies the following parameters :

    timeout no-session: No-session timeout for the tunnel. The default value is 0 seconds.

    timeout *timeout_value*: Timeout value in seconds. The valid values range from 1 to 86400 seconds.
  - **timeout busy** *value*: Specifies the timeout period before marking an LNS server as busy. The valid values range from 60 to 65535 seconds.

- **vrf** *vrf_name*: Specifies the Virtual routing and forwarding (VRF) name of the tunnel.

Monitoring and troubleshooting

This section describes key operational commands available for diagnosing tunnel states and analyzing tunnel failures.

View L2TP tunnel instance details

Display detailed information about a specific L2TP tunnel instance for troubleshooting or monitoring.

Use this CLI command to inspect the operational state and key parameters of an L2TP tunnel. This is especially useful for diagnosing tunnel connectivity, performance, or configuration issues.

Procedure

Step 1 Use the **show l2tp-tunnel-instance upf** *upf-name* **tunnel-id** *value* command to display the specific L2TP tunnel instance details.

Example:

```
show l2tp-tunnel-instance upf asr9k-lac tunnel-id 13176

Tue Oct  21 11:38:08.584 UTC+00:00
result
[
  {
    "Tunnel-ID": 13176,
    "Remote-Tunnel-ID": 8858,
    "Tunnel-Type": "LAC",
    "UP": "asr9k-1",
    "FSM-state": "established",
    "Src-Endpoint": "10.1.39.139",
    "Dst-Endpoint": "10.1.34.52",
    "Group-ID": "TWO",
    "Control-NS": 4,
    "Control-NR": 2,
    "Remote-RWS": 512,
    "Disconnect-state": "None",
    "Clear-sessions": "None",
    "Delete-state": "None",
    "Recovery-state": "Initial"
  }
]
```

Step 2 (Optional) Use the **show l2tp-tunnel-instance upf** *upf-name* command to display detailed information about active L2TP tunnel instances associated with the UPF.

Example:

```
show l2tp-tunnel-instance upf asr9k-1
Tue Oct  21 11:36:54.595 UTC+00:00
result
[
  {
    "Tunnel-ID": 13176,
    "Remote-Tunnel-ID": 8858,
    "Tunnel-Type": "LAC",
    "UP": "asr9k-1",
```

```
    "FSM-state": "established",
    "Src-Endpoint": "10.1.39.139",
    "Dst-Endpoint": "10.1.34.52",
    "Group-ID": "TWO",
    "Control-NS": 4,
    "Control-NR": 2,
    "Remote-RWS": 512,
    "Disconnect-state": "None",
    "Clear-sessions": "None",
    "Delete-state": "None",
    "Recovery-state": "Initial"
  },
  {
    "Tunnel-ID": 1391,
    "Remote-Tunnel-ID": 8859,
    "Tunnel-Type": "LAC",
    "UP": "asr9k-1",
    "FSM-state": "established",
    "Src-Endpoint": "10.1.39.139",
    "Dst-Endpoint": "10.1.34.52",
    "Group-ID": "ONE",
    "Control-NS": 6,
    "Control-NR": 3,
    "Remote-RWS": 512,
    "Disconnect-state": "None",
    "Clear-sessions": "None",
    "Delete-state": "None",
    "Recovery-state": "Initial"
  }
]
```


View detailed LNS session information using the sublabel identifier

Enable network administrators to retrieve and analyze comprehensive debug-level information for a specific LNS subscriber session using the session's sublabel.

Procedure

Use the **show subscriber lns sublabel** *sublabel* **debug** command to view detailed debug information for a specific LNS subscriber session.

Example:

```
bng# show subscriber lns sublabel 16777219 debug

Mon Oct  20 06:30:27.240 UTC+00:00
subscriber-details
{
  "subResponses": [
    {
      "state": "complete",
      "key": {
        "routerID": "asr9k-1",
        "portID": "GigabitEthernet0/1/0/1",
        "subLabel": "16777218",
        "upSubID": "1"
      },
```

```
      "flags": [
        "SM_START_DONE",
        "SM_ACTIVATE_DONE",
        "SM_UPDATE_DONE",
        "IPCP_UP",
        "IPV6CP_UP"
      ],
      "lcpInfo": {
        "state": "opened",
        "keepAliveInterval": 60,
        "keepAliveRetries": 5,
        "localMru": 1492,
        "peerMru": 1492,
        "localMagic": "0xe2ab84f",
        "peerMagic": "0xe2ab850",
        "authOption": "PAP",
        "authCompleted": true,
        "username": "cnbng"
      },
      "ipcpInfo": {
        "state": "opened",
        "peerIpv4Pool": "pool-ISP",
        "peerIpv4Address": "11.0.32.2",
        "peerIpv4Netmask": 22,
        "localIpv4Address": "11.0.32.1",
        "isIpamPoolIPaddr": true
      },
      "ipv6cpInfo": {
        "state": "opened",
        "localIntfID": "0x1",
        "peerIntfID": "0xcc11000000010001"
      },
      "lnsInfo": {
        "srcIP": "10.1.39.139",
        "dstIP": "10.1.34.52",
        "state": "established",
        "profileName": "l2tp-prof2",
        "tunnelClientAuthID": "bng-lac",
        "tunnelServerAuthID": "Local-DC",
        "callSerialNumber": 16777408,
        "localTunnelID": 12226,
        "localSessionID": 11428,
        "remoteTunnelID": 1017,
        "remoteSessionID": 24885
      },
      "sessionType": "lns",
      "vrf": "default",
      "AuditId": 4,
      "sess-events": [
        "Time, Event, Status",
        "2025-10-20 06:30:01.966696374 +0000 UTC, Sublabel Alloc, complete",
        "2025-10-20 06:30:01.971563927 +0000 UTC, L2TP-ID Alloc, complete",
        "2025-10-20 06:30:01.986624734 +0000 UTC, UP Interface Create, complete",
        "2025-10-20 06:30:02.010855854 +0000 UTC, SM Session-Start, complete",
        "2025-10-20 06:30:02.038466759 +0000 UTC, L2TP session, complete",
        "2025-10-20 06:30:02.038504381 +0000 UTC, LCP, up",
        "2025-10-20 06:30:02.07193459 +0000 UTC, SM Session-Activate, complete",
        "2025-10-20 06:30:02.102904338 +0000 UTC, IPv6CP, up",
        "2025-10-20 06:30:03.202555529 +0000 UTC, IPCP, up"
      ]
    }
  ]
}
```

This information helps network administrators verify session parameters and troubleshoot LNS connectivity issues.

View L2TP tunnel disconnect history

Retrieve the recent disconnect history for L2TP tunnels on a specified UPF.

Use this task to investigate tunnel stability, common disconnect reasons, or repeated failures.

Procedure

Step 1 Use the **show disconnect-history l2tp-tunnel upf** *upf-name* command to display all recent disconnects for a UPF.

Example:

```
bng# show l2tp-tunnel disconnect-history upf asr9k-lns

Wed Jan  21 04:57:05.678 UTC+00:00
-------------------------------------------------------------------------------
[Disconnect Reason]                    [Last Disconnect Time]       [Tunnel-id]
-------------------------------------------------------------------------------
HELLORetransFailure                    2026/01/21 04:54:22.406      5951
-------------------------------------------------------------------------------
```

Step 2 Use the **show disconnect-history l2tp-tunnel upf** *upf-name* **unique** command to display disconnect records summarized by disconnect reason. For each reason, it shows the disconnect time and tunnel ID of the most recent event, along with a count of how many disconnects occurred for that reason.

Example:

```
bng# show disconnect-history l2tp-tunnel upf asr9k-lns unique

Wed Jan  21 04:56:32.195 UTC+00:00
-----------------------------------------------------------------------------------------------
[Disconnect Reason]                [Last Disconnect Time]       [Tunnel-id]      [Count]
-----------------------------------------------------------------------------------------------
HELLORetransFailure                2026/01/21 04:54:22.406      5951             1
L2TP Admin Delete                  2026/01/21 04:46:17.713      34975            2
No-Session Timeout                 2026/01/21 04:45:32.125      2028             1
-----------------------------------------------------------------------------------------------
```

Step 3 Optionally, use the **show disconnect-history l2tp-tunnel upf** *upf-name* **last** command to display up to the last 1000 disconnect records.

Example:

```
bng# show disconnect-history l2tp-tunnel upf asr9k-lns last

Wed Jan  21 04:56:56.831 UTC+00:00
-----------------------------------------------------------------------------------------------
[Disconnect Reason]                    [Last Disconnect Time]       [Tunnel-id]      [Count]
-----------------------------------------------------------------------------------------------
HELLORetransFailure                    2026/01/21 04:54:22.406      5951             1
L2TP Admin Delete                      2026/01/21 04:46:17.713      34975            1
No-Session Timeout                     2026/01/21 04:45:32.125      2028             1
L2TP Admin Delete                      2026/01/21 04:44:14.714      43738            1
-----------------------------------------------------------------------------------------------
```

Step 4 Optionally, use the **show disconnect-history l2tp-tunnel upf** *upf-name* **details** command to display last 100 disconnect records along with tunnel details.

Example:

```
bng# show disconnect-history l2tp-tunnel upf asr9k-lns details

Wed Jan  21 04:57:21.810 UTC+00:00
{
  "subResponses": [
    {
      "LocalTunnelId": 43738,
      "RemoteTunnelId": 22331,
      "Upf": "asr9k-lns",
      "SrcIp": "10.1.12.103",
      "DstIp": "10.1.15.136",
      "GroupId": "one",
      "TunnelType": "LAC",
      "DiscReason": "L2TP Admin Delete"
    },
    {
      "LocalTunnelId": 2028,
      "RemoteTunnelId": 22332,
      "Upf": "asr9k-lns",
      "SrcIp": "10.1.12.103",
      "DstIp": "10.1.15.136",
      "GroupId": "one",
      "TunnelType": "LAC",
      "DiscReason": "No-Session Timeout"
    },
    ...
    {
```

Step 5 Optionally, use the **show disconnect-history l2tp-tunnel upf** *upf-name* **filter tunnel-id** *tunnel-ID* command to display disconnect history details for tunnels that match the specified tunnel ID and UPF name.

Example:

```
bng# show disconnect-history l2tp-tunnel upf asr9k-lns filter tunnel-id 2028

Wed Jan  21 04:57:39.563 UTC+00:00
filter tunnel-id 2028
{
  "subResponses": [
    {
      "LocalTunnelId": 2028,
      "RemoteTunnelId": 22332,
      "Upf": "asr9k-lns",
      "SrcIp": "10.1.12.103",
      "DstIp": "10.1.15.136",
      "GroupId": "one",
      "TunnelType": "LAC",
      "DiscReason": "No-Session Timeout"
    }
  ]
}
```

Note A maximum of 30,000 L2TP-tunnel disconnect history records can be stored across all UPFs and instances (this is the system-wide limit for the entire Control Plane). Once the limit is reached, the oldest record is overwritten as new records are added.

Clear L2TP-tunnel disconnect history records

Remove L2TP-tunnel disconnect history records from the system for maintenance or troubleshooting.

Before you begin

- Confirm which records (all, or specific User Plane) you wish to clear.

Procedure

Step 1 Use the **clear disconnect-history l2tp-tunnel** command to clear all L2TP-tunnel disconnect history records.

Step 2 Use the **clear disconnect-history l2tp-tunnel** *user-plane-name* command to clear records for a specific User Plane.