

PPPoE Subscriber Management

- Feature Summary and Revision History, on page 1
- Feature Description, on page 2
- Configuring the PPPoE Subscriber Management Feature, on page 10
- View PPPoE or LNS session event history, on page 13
- Stateless Address Autoconfiguration (SLAAC) , on page 16

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 the **show subscriber pppoe [lns] sublabel** *sublabel* **debug** command for viewing event history details for a specific PPPoE or LNS session.	2026.01.0
Introduced support for PPPoE session limit.	2025.02.0

Revision Details	Release
Introduced support for Stateless Address Autoconfiguration (SLAAC).	2024.04.0
Enhancement Introduced: The PPPoE is NSO-integrated.	2021.04.0
First introduced.	2021.01.0

Feature Description

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

Point-to-Point Protocol (PPP) over Ethernet (PPPoE) is a point-to-point link with the subscriber over an Ethernet network where the standard PPP negotiations are used for authentication and IPv4 address assignment. The basic PPPoE is defined in RFC-2516. This RFC defines two distinct stages:

- **Discovery stage:** This sets up a point-to-point session over which PPP can run between two points. For example, between the CPE and Broadband Network Gateway (BNG). This is the PPPoE protocol itself.

  Unlike PPP, the PPPoE discovery protocol defines a client-server relationship with the client initiating the discovery of the server and the subsequent setup of the point-to-point link.

- **Session stage:** This runs over the established point-to-point connection, negotiating the PPP protocols (LCP, Authentication, IPCP) as required for a standard PPP interface.

  The session stage carries the data packets from the PPPoE (this includes PPP protocol negotiation) and the actual data packets to and from the subscriber.

PPPoE Overview

The cnBNG CP supports the standard PPPoE protocol, as defined in RFC-2516. It implements the PPPoE server functionality, that is, providing PPPoE sessions to subscribers who request them. More specifically, it supports the following functionality:

- Handling incoming PPPoE Active Discovery Initiation (PADI) packets and replying with a PPPoE Active Discovery Offer (PADO) packet when the PADI is valid.
- Handling incoming PPPoE Active Discovery Request (PADR) packets and setting up a PPPoE session for the subscriber when the PADR is valid. It also replies with a PPPoE Active Discovery Session (PADS) with an allocated session-id. When the PADR is not valid (or session setup fails), a PADS is sent containing a zero session-id and an error tag.
- Handling incoming PPPoE Active Discovery Termination (PADT) packets and terminating the corresponding PPPoE sessions.
- Sending a PADT packet to the subscriber when terminating a PPPoE session.

PPPoE Features

The cnBNG supports the following PPPoE features.

PPPoE Tag Support

cnBNG supports the following PPPoE tags as defined in RFC-2516.

- Service-Name
- AC-Name tag
- AC-Cookie
- Host-Uniq tag
- Relay-Session-Id tag
- End-Of-List tag
- Vendor-Specific tags
- Error tags
- Max-payload tag

Interface types

PPPoE is generally supported on all types of Ethernet interfaces. The cloud-native CP supports PPPoE if the configuration is present either on the port identifier, NAS level, or at the router level. The UP is responsible for the interfaces where the PPPoE punt inject towards CP can be enabled.

CoS Bits

The cnBNG allows configuration of the Class-of-Service (CoS) bits value used in the Ethernet header of PADx packets. This ensure that the PPPoE control packets get treated at a higher priority. The cnBNG CP passes these values in the inject packet and the UP places these CoS values in the PADx packets it forwards towards the CPE.

Service Selection

The PPPoE Service Selection feature uses service tags to enable a PPPoE server to offer PPPoE clients a selection of different services in the PADO. Then the client chooses one of the services offered and then sends the desired service name in a PADR. This feature enables service providers to offer a variety of services and to charge customers according to the chosen services.

Whenever a PADI is received containing one of the locally configured service-names, the PADO response contains all the configured service-names.

A configuration is also provided to allow the user to disable Service Selection. In this case, the PADO only contains the service-name that was in the original PADI.

PPPoE Session Limits

The PPPoE session limits feature enables you to control the number of PPPoE sessions that can be established on a specific User Plane (UP) and PPPoE profile. By restricting session creation, this feature protects cnBNG system resources when multiple subscribers attempt to access broadband services simultaneously.

This feature provides enhanced configuration flexibility by allowing you to set session limits based on various parameters, including:

- **circuit-id** – Maximum number of sessions allowed per Circuit-ID.
- **circuit-id-and-remote-id** – Maximum number of sessions allowed per combination of Circuit-ID and Remote ID.
- **mac** – Maximum number of sessions allowed per MAC address.
- **max** – Maximum number of sessions allowed under the PPPoE profile.
- **outer-vlan** – Maximum number of sessions allowed per outer VLAN, per access interface.

PPPoE session limit based on circuit-id and remote-id

Table 3: Feature History

Feature Name	Release Information	Description
PPPoE session limit based on circuit-id and remote-id	2025.02.0	You can now limit the number of PPPoE sessions on a designated UP and PPPoE profile, using the combination of Circuit-ID and Remote-ID as filtering criteria. This feature ensures optimal protection of cnBNG system resources, effectively handling simultaneous session requests from multiple subscribers.

This feature restricts the number of PPPoE sessions on a specific UP and PPPoE profile, using the combination of Circuit-ID and Remote-ID as the enforcement criteria. If the predefined session limit is exceeded, cnBNG automatically rejects new subscriber session requests associated with that Circuit-ID and Remote-ID.

Benefits of PPPoE session limit

- **Resource protection**: By controlling the number of active sessions, you can protect cnBNG system resources from potential strain due to multiple connection attempts.
- **Enhanced stability:** The feature contributes to maintaining system stability and performance, particularly during peak usage times when multiple subscribers are attempting to connect.

Configure PPPoE session limits

Follow these steps to configure PPPoE session limits.

Procedure

Step 1 Configure PPPoE session limits to control the maximum number of PPPoE sessions that can be established on a cnBNG router.

Example:

```
config
  profile pppoe pppoe_profile_name
```

```
session-limit { circuit-id | circuit-id-and-remote-id | mac | max | outer-vlan
} count
exit
```

This is a sample configuration.

```
config
   profile pppoe prof1
     session-limit circuit-id-and-remote-id 5
   exit
```

Step 2 Use the **show subscriber pppoe { count | detail }** command to verify the configuration.

Example:

```
bng# show subscriber pppoe count

Tue Oct  21 11:36:40.189 UTC+00:00
subscriber-details
{
  "sessionCount": 3
}
```

PPP Overview

The Point-to-Point Protocol provides a standard method for transporting multiprotocol datagrams over point-to-point links. It defines an encapsulation scheme, a link layer control protocol (LCP) and a set of network control protocols (NCPs) for different network protocols that can be transmitted over the PPP link.

The LCP is used to configure and maintain the data link. PPP peers use the LCP to negotiate various link layer properties or characteristics.

An NCP is used to establish and configure the associated network protocol before data packets for the protocol are transmitted. For example, IP Control Protocol (IPCP) is used to negotiate IPv4 addresses between peers.

Between LCP and NCP negotiation phases there is an optional authentication phase that the LCP exchanges are agreed upon. Several different authentication schemes are selected with Challenge Handshake Authentication Protocol (CHAP) being the most prevalent one. The basic PPP protocol is defined in RFC 1661 and there are extensions to it for various features.

PPP Features

The cnBNG supports the following point-to-point protocols required for bringing up a PPPoE session.

- Link Control Protocol (LCP): This is used for PPP link configuration.
- IP Control Protocol (IPCP): This is used to negotiate IPv4 addresses between peers.
- IPv6 Control Protocol (IPv6CP): This is used to negotiate IPv6 interface ID.
- Password Authentication Protocol (PAP): This is used to verify the identity of the peer by means of a two-way handshake
- Challenge Handshake Authentication Protocol (CHAP): This is used to verify the identity of the peer by means of a three-way handshake.

For more information about the protocols and their negotiation, refer the respective RFCs.

Address Assignment Strategies

The IPv4 address assignment occurs as part of the IPCP negotiation. The address can be part of the RADIUS profile. Often it is the RADIUS profile that specifies the pool to use and the Control Plane (CP) selects an address from that pool. If neither the address nor pool comes from the RADIUS, the PPP profile configuration (on the box) specifies which pool name to use. This profile is attached to the port identifier where the PPP packets are received.

The IPv6 address assignment occurs in two phases:

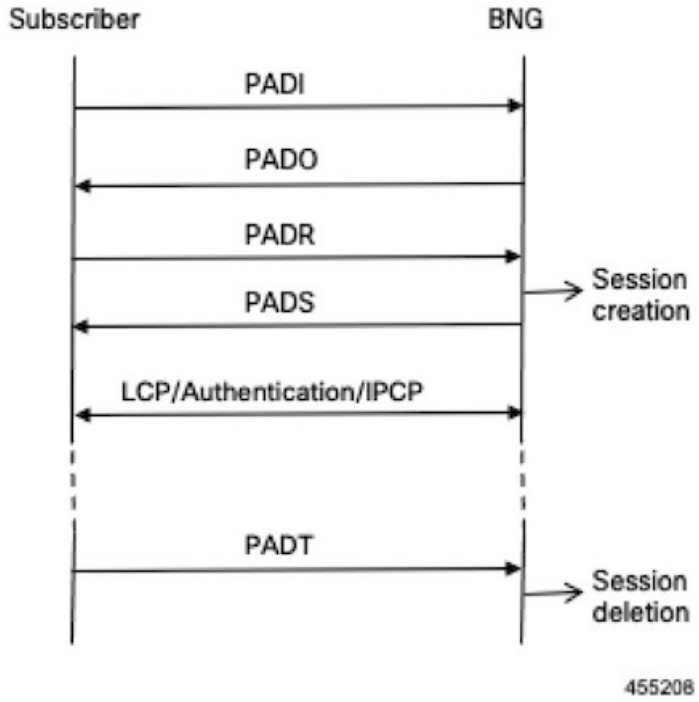
- First, as part of the IPv6CP, the interface-ID is negotiated with the CPE, which is used for link local negotiation.
- Second, after the CPE initiates the DHCPv6 protocol to get IPV6 IANA or IAPD (or both) address allocation, it gets the IPv6 address from either the RADIUS or from a pool.

How it Works

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

PPPoE Handling

The PPPoE discovery-stage protocol consists of basic packet exchange between the subscriber and server (cnBNG). The following illustration displays the flow of events.



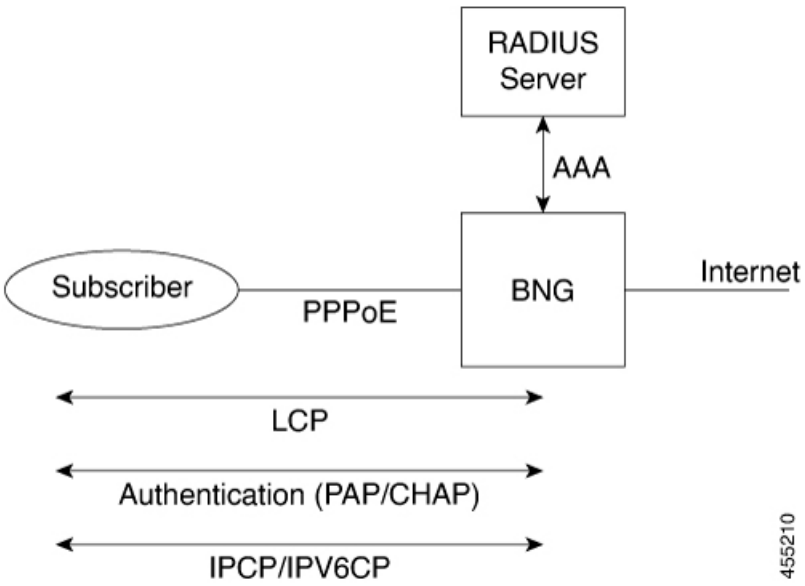
In brief, the protocol can be summarized as follows:

- When the subscriber wishes to establish a PPPoE session, it sends PADI message to the server.
  - The PADI may be multicast, if the subscriber tries to find out if any servers are available.
  - The PADI contains a Service-Name tag, which indicates the service that it wants the server to provide.
- When a server receives a PADI message, it checks if it can provide the service requested to the subscriber. If it can, it replies with a PADO message.
  - The PADO message is unicast to the peer. It contains the Service-Name the client requested.
- When the subscriber receives the PADO messages from the servers, it selects the server to connect to and sends a PADR message.
  - The PADR message is unicast, directed to the specific server with which it wants to establish a session.
  - The PADR message also contains the Service-Name tag.
- When the server receives a PADR message, it checks if it can provide the service to the subscriber.
  - If it can, it chooses a 16-bit Session-Id to identify the session of the subscriber and sets up the necessary state for the subscriber. It then replies with a PADS confirmation, which contains the Session-Id to indicate to the subscriber that the session is established.
  - If it cannot provide a session, it replies with a PADS containing an Error-tag, which indicates the reason it cannot. This PADS contains a zero Session-id.
- After the PADS is sent, the subscriber and server negotiate PPP in the standard way.
- When either the subscriber or the server wants to terminate the session, it sends PADT message to the peer with the Session-Id. This clears up all the states associated with the session.

  This completes the PPPoE discovery stage. the peers can now start the PPP negotiation.

PPP Handling

The network topology of the PPP is the point-to-point link between the BNG and the subscriber (this link is established during the PPPoE Discovery phase):



The PPPoE subscriber is viewed like any other PPP peer – LCP, Authentication and IPv4CP or IPv6CP (or both) are negotiated to establish the PPP link.

The standard scenario where the BNG terminates both the PPPoE and PPP subscriber session is referred to as PPP Termination and Aggregation (PTA). This distinguishes it from the more complex L2TP Access Concentrator (LAC) and L2TP Network Server (LNS) scenarios where the PPPoE is terminated locally on the BNG but the PPP session is terminated on a separate node from over L2TP to an upstream box known as an LNS.

Call Flows

This section includes the following high-level call flow.

PPPoE Bring-Up Call Flow

In cnBNG, the PPPoE and PPP Control Plane runs the overall PTA session bring-up, which includes the PPPoE and PPP negotiation as shown in the following call-flow.

Figure 1: cnBNG PPPoE Bring-Up Call Flow

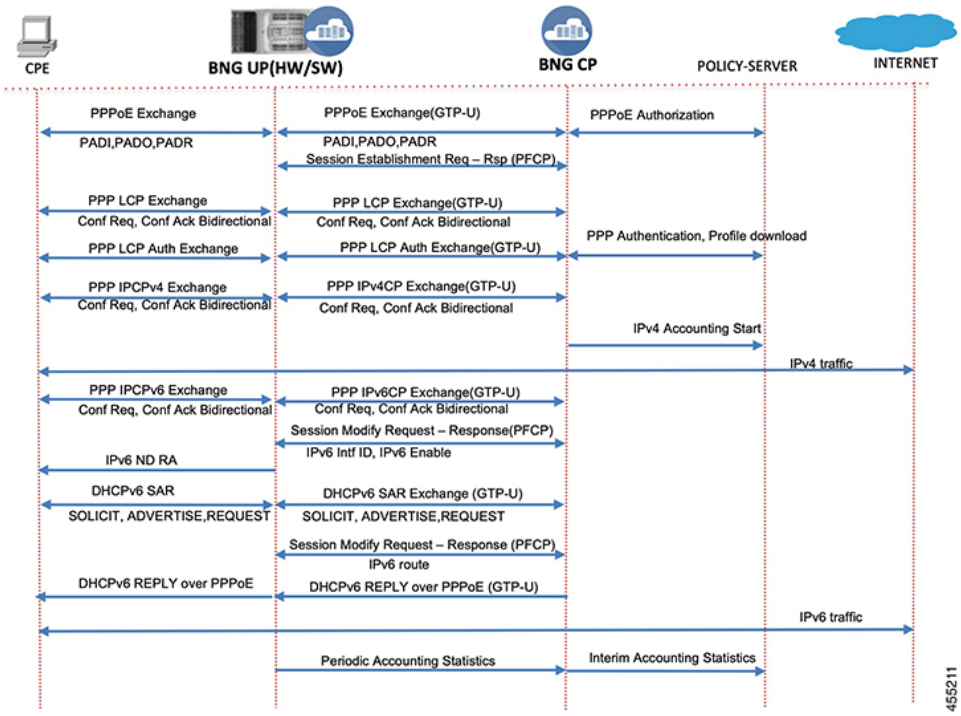


Table 4: cnBNG PPPoE Call Flow Description

Steps	Description
1	The CPE initiates the PPPoE packet exchange. The BNG-CP verifies the tags, session limits, service name, and so on and creates a PPPoE interface.
2	The BNG-CP sends a Success PADS message with an assigned PPPoE session ID.
3	The CPE and BNG-CP negotiate the LCP link parameters and authorization methods to use.
4	The BNG-CP authenticates the CPE with the provided username and password via AAA and downloads the network level parameters.
5	The CPE starts the IPv4CP and gets the IPv4 address. The BNG-CP programs the IPv4 route and features on the BNG-UP. Accounting start is initiated for IPv4.
6	Now bidirectional IPv4 traffic is enabled for the subscriber with the applied features.
7	Optionally, the CPE starts IPv6CP in case of dual stack. The local and peer interface ID are negotiated and the BNG-UP is programmed to allow link local negotiation to occur.
8	The BNG-UP completes the link local addressing with the IPv6 ND router advertisement.

Steps	Description
9	The CPE starts the DHCPv6 packet exchange on the negotiated PPPoE session to get the global IPv6 address assignment.
10	The BNG-CP programs the IPv6 routes and features into the BNG-UP and responds to the CPE with the DHCPv6 Reply packet to acknowledge that the IPv6 is up. At this stage, the session is converted into a dual stack in the CP.
11	The subscriber can now send and receive IPv6 traffic from the Internet.
12	The BNG-UP collects and pushes the interim statistics to the BNG-CP. The BNG-CP pushes these statistics to the Policy Plane for billing.

Standard Compliance

The PPPoE Subscriber Management feature is aligned with the following standards:

- RFC 1661 Point-to-Point Protocol
- RFC 2516. A Method for Transmitting PPP Over Ethernet (PPPoE)

Limitations

The PPPoE Subscriber Management feature has the following limitations:

- Session throttling is not supported
- The PPPoE profile and PPP feature template configuration changes are applied only to the new sessions. These changes are not applied to the existing sessions.
- Update of PPP features via CoA is not supported.
- In deployment, it is not expected that one address family interface (AFI) of session goes down and another AFI bring-up is inflight. This is not supported. For example, the following sequence is not supported.
  1. Dual-stack is UP.
  2. IPv4 release is received (Session becomes IPv6 only).
  3. IPv4 Discover is received (Session is IPv6 only but IPv4 is inflight).
  4. IPv6 release is received (Session becomes IPv4 is inflight).
  5. IPv4 Request is receive (Session is IPv4 only).

     In Step 3, IPv4 session is inflight and in step 4, IPv6 is going down, and in step 5, IPv4 session bring-up is complete. Such sequence of events is not expected in deployment and is not supported.

Configuring the PPPoE Subscriber Management Feature

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

Configuring the PPPoE Subscriber Management feature involves the following steps:

1. Creating the PPPoE profile
2. Creating the PPP Feature template

Creating PPPoE Profile

Use the following commands to create a PPPoE profile and provide the PPPoE protocol specific parameters.

```
config
  profile pppoe pppoe_profile_name
  mtu mtu
  service-selection-disable [ true | false ]
  max-payload minimum { payload_value } maximum { payload_value }
  service-name service_name
  ac-name ac_name
  ac-cookie ac-cookie_name
  session max limit { count } threshold { count }
  session mac limit { count } threshold { count }
  session circuit-id limit { count } threshold { count }
  session outer-vlan limit { count } threshold { count }
  timeout-completion period
  control-packets priority cos_value
  exit
```

NOTES:

- **profile pppoe** *pppoe_profile_name*: Specifies the PPPoE profile name.
- **mtu** *mtu*: Specifies the default PPP maximum transmission unit (MTU) value to use if the Max-Payload tag is not provided. The valid values range from 500 to 2000. The default value is 1492.
- **service-selection-disable [true | false]**: Enables or disables the advertising of extra service names in the PADO packets. True enables the service and false disables the service. The default value is false.
- **max-payload minimum {** *payload_value* **} maximum {** *payload_value* **}**: Specifies the supported PPPoE service name. Multiple service names can be configured simultaneously. The valid value is an alphanumeric string ranging from 1 to 256. All service names are accepted.
- **service-name** *service_name*: Specifies the supported PPPoE service name. Multiple service names can be configured simultaneously. The valid value is an alphanumeric string ranging from 1 to 256. All service names are accepted.
- **ac-name** *ac_name*: Specifies the access concentrator (AC) to use in the PADO packets. The valid value is an alphanumeric string ranging from 1 to 256. The default ac-name is the router hostname.
- **ac-cookie** *ac-cookie_name*: Specifies the AC-Cookie to use in the PADO packets. The valid value is an alphanumeric string ranging from 1 to 256.
- **session max limit {** *count* **} threshold {** *count* **}**: Specifies the total maximum number of sessions and threshold allowed per User Plane per profile. The valid values range from 1 to 65535. The default value is 65535.
- **session mac limit {** *count* **} threshold {** *count* **}**: Specifies the maximum number of sessions and threshold allowed per UP per peer profile. The valid values range from 1 to 65535. The default value is 65535.

  When the threshold is passed, a syslog is printed as a warning.

- **session circuit-id limit {** *count* **} threshold {** *count* **}**: Specifies the maximum number of sessions and threshold allowed per circuit-id. The valid values range from 1 to 65535. The default value is 65535.

  When the threshold is passed, a syslog is printed as a warning.

- **session outer-vlan limit {** *count* **} threshold {** *count* **}**: Specifies the maximum number of sessions and threshold allowed per UP per peer profile. The valid values range from 1 to 65535. The default value is 65535.

  When the threshold is passed, a syslog is printed as a warning.

- **timeout-completion** *period*: Specifies the maximum time to wait for the session to be completed (an NCP to come up for PTA sessions or the L2TP tunnel to be setup for LAC sessions) before terminating the session. The valid values range from 30 to 600 seconds. The default value is 120 seconds.

- **control-packets priority** *cos_value*: Specifies the CoS to use in the PADx packets. The valid values range from 0 to 7. The default CoS bits are used.

Creating the PPP Feature Template

Use the following commands to create a PPP feature template.



Note The PPP feature template allows per subscriber PPP parameters.

```
config
  profile feature-template feature_template_name
  ppp
    authentication { chap  | pap }
    chap hostname chap_hostname
    chap password chap_password
    ipcp dns ipv4_address
    ipcp peer-address-pool ipam_pool_name
    ipcp renegotiation ignore
    ipcp wins ipv4_address
    ipcpv6 renegotiation ignore
    ipcp wins ipv4_address
    max-bad-auth count
    max-configure count
    max-failure count
    pap accept-null-password
    timeout absolute seconds
    timeout authentication seconds
    timeout retry seconds
    keepalive interval secondsretryseconds[ disable ]
    exit
```

NOTES:

- **profile feature-template** *feature_template_name*: Specifies the profile feature template name.

- **ppp**: Enters the PPP Configuration mode to configure the PPP feature.

- **authentication { chap | pap }**: Specifies the authentication type as CHAP or PAP.
- **chap hostname** *chap_hostname*: Specifies the hostname to use for CHAP authentication. The valid values range from 1 to 64. The default value is the router hostname.
- **chap password** *chap_password*: Specifies the password to use for CHAP authentication.
- **ipcp dns** *ipv4_address*: Specifies the DNS address to use for the peer.
- **ipcp peer-address-pool** *ipam_pool_name*: Specifies the address pool to use to obtain an IPv4 address for the peer.
- **ipcp renegotiation ignore**: Specifies to ignore the attempts of the peer to renegotiate IPCP. The entire PPPoE session is terminated on renegotiation.
- **ipcp wins** *ipv4_address*: Specifies the Windows Internet Name Service (WINS) address to use for the peer.
- **max-bad-auth** *count*: Specifies the maximum authentication failures to allow. The valid values range from 0 to 10. The default value is 0.
- **max-configure** *count*: Specifies the maximum number of Conf-Reqs to send without a response. The valid values range from 4 to 20. The default value is 10.
- **max-failure** *count*: Specifies the maximum number of Conf-Naks to send. The valid values range from 2 to 10. The default value is 5.
- **pap accept-null-password**:Accepts the null password feature for PAP.
- **max-failure** *count*: Specifies the maximum number of Conf-Naks to send. The valid values range from 2 to 10. The default value is 5.
- **timeout absolute** *seconds*: Specifies the absolute timeout for a PPP session. The valid values range from 0 to 70000000 minutes.
- **timeout authentication** *seconds*: Specifies the total time to allow for authentication to complete. The valid values range from 3 to 30 seconds. The default value is 10.
- **timeout retry** *seconds*: Specifies the maximum time to wait for a response to a Conf-Req. The valid values range from 1 to 10 seconds. The default value is 3.
- **keepalive interval** *seconds* **retry** *seconds* **[disable]**: Specifies the keepalive interval and the retry attempts for the subscribers. The valid values range from 10 to 120 seconds for the keepalive interval. The default is 60 seconds. The valid values range from 1 to 255 for the retry attempt. The default value is 5 counts.

View PPPoE or LNS session event history

Examine the recent event history for a specific PPPoE or LNS session to assist with troubleshooting and monitoring.

Procedure

Step 1 Use the **show subscriber pppoe [lns] sublabel** *sublabel* **debug** command to view detailed debug information for a specific PPPoE subscriber session.

Example:

```
bng# show subscriber pppoe sublabel 16777219 debug

Fri Oct  17 06:12:26.852 UTC+00:00
subscriber-details
{
  "subResponses": [
    {
      "state": "complete",
      "key": {
        "routerID": "asr9k-1",
        "portID": "GigabitEthernet0/0/0/1",
        "outerVlan": 100,
        "innerVlan": 200,
        "macAddr": "cc11.0000.0001",
        "pppoesessionID": 3,
        "subLabel": "16777219",
        "upSubID": "3",
        "HostUniq": "0x686f7374756e71",
        "circuitID": "circuit1",
        "remoteID": "remote1"
      },
      "flags": [
        "SM_START_DONE",
        "SM_ACTIVATE_DONE",
        "SM_UPDATE_DONE",
        "PPPOE_UP_DONE",
        "IPCP_UP",
        "IPV6CP_UP"
      ],
      "pppoeInfo": {
        "profileName": "pppoe-prof1",
        "mtu": 1500
      },
      "lcpInfo": {
        "state": "opened",
        "keepAliveInterval": 60,
        "keepAliveRetries": 5,
        "localMru": 1500,
        "peerMru": 1500,
        "localMagic": "0xf936c66e",
        "peerMagic": "0x112233",
        "authOption": "PAP",
        "authCompleted": true,
        "username": "cnbng"
      },
      "ipcpInfo": {
        "state": "opened",
        "peerIpv4Pool": "pool-ISP",
        "peerIpv4Address": "11.0.160.3",
        "peerIpv4Netmask": 22,
        "localIpv4Address": "11.0.160.1",
        "isIpamPoolIPaddr": true
      },
      "ipv6cpInfo": {
```

```
        "state": "opened",
        "localIntfID": "0x1",
        "peerIntfID": "0xcc11000000010000"
      },
      "sessionType": "pta",
      "vrf": "default",
      "AuditId": 4,
      "sess-events": [
        "Time, Event, Status",
        "2025-10-17 06:12:14.933636323 +0000 UTC, Session-ID Alloc, complete",
        "2025-10-17 06:12:14.94356763 +0000 UTC, UP Interface Create, complete",
        "2025-10-17 06:12:14.963876513 +0000 UTC, SM Session-Start, complete",
        "2025-10-17 06:12:15.978631055 +0000 UTC, LCP, up",
        "2025-10-17 06:12:16.016685284 +0000 UTC, SM Session-Activate, complete",
        "2025-10-17 06:12:16.06998785 +0000 UTC, IPCP, up",
        "2025-10-17 06:12:16.094605971 +0000 UTC, IPv6CP, up"
      ]
    }
  ]
}
```


```
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"
      ]
    }
  ]
}
```


Step 2 (Optional) Use the **subscriber event-trace-disable** command to disable event tracing.

Step 3 Use the **subscriber event-trace-max-count** *number* to set or adjust the maximum number of events retained in the history log for sessions.

Stateless Address Autoconfiguration (SLAAC)

Table 5: Feature History

Feature Name	Release Information	Description
Stateless Address Autoconfiguration (SLAAC)	2024.04.0	This feature allows each IPv6 host to generate its own address using local and router-advertised information, simplifying the integration of new IPv6 hosts without extensive configuration.

Stateless Address Autoconfiguration (SLAAC) allows each IPv6 host in a network to generate their own address using a combination of local and router-advertised information. This process is lightweight and does not require a server to track assigned addresses or their states.

Defined in RFC 4862, SLAAC simplifies the integration of new IPv6 hosts into a network without extensive configuration. It also facilitates the migration of existing IPv4 hosts to IPv6 networks.

This DHCPv6 protocol is a stateful counterpart to IPv6 SLAAC, and can be used separately, or concurrently with SLAAC, to obtain configuration parameters.

IPv6 Address Assignment

The Customer Premises Equipment (CPE) or Residential Gateway (RG) is assigned a /64 IPv6 prefix from the Broadband Network Gateway (BNG). Using this prefix, the CPE/RG generates its 128-bit global IPv6 address for the WAN side link.

Configuration Flags Usage

SLAAC relies on the ICMPv6 protocol to advertise the IPv6 prefix to hosts. When an IPv6-enabled host is detected in the network, the IPv6 router sends an ICMPv6 Router Advertisement (ICMPv6.Type = 134) packet. This packet contains the /64 prefix and other necessary information such as MTU and MAC address of the interface. It also includes specific flags relevant to the overall IPv6 enablement of the host:

- **Managed Address Configuration Flag (M flag)**

  When set to 1 (on), the router instructs the host to use a stateful DHCPv6 server for its global unicast address and all other addressing information. If the M flag is set, the O flag can be ignored because the DHCPv6 server returns all available information.

- **Other Configuration Flag (O flag)**

  When set to 1 (on), the DNS information is obtained from a stateless DHCPv6 server. The router instructs the nodes to auto-configure an address using SLAAC and to request DNS information from the DHCPv6 server.

- If neither the M flag nor the O flag is set, this indicates that no DHCPv6 server is available on the segment.

The M and O flags provide the host with a comprehensive view of how it can enable its IPv6 configuration. This includes prefix and DNS information, using SLAAC protocol. This flexibility ensures that hosts can be seamlessly integrated into the IPv6 network with minimal configuration effort.

Configure SLAAC for PPPOE

Procedure

Step 1 Use the following sample configuration to configure SLAAC for PPPOE.

Example:

```
ipam
   instance 1
      source local
      address-pool pool-ipv6
          vrf-name abc
          ipv6
             prefix-ranges
                 split-size
                    per-cache 512
                    per-dp    512
                 exit
             prefix-range 2001:DB8:: length 48
          exit
```

```
      exit
   exit
exit
profile slaac slaac-prof1
    managed-config-flag enable
    other-config-flag enable
    prefix-pool pool-ipv6
exit
profile subscriber subs-prof1
    pppoe-profile pppoe-prof1
    slaac-profile slaac-prof1
exit
user-plane
    instance 1
       user-plane asr9k-1
          peer-address ipv4 10.1.1.1
          subscriber-profile subs-prof1
       exit
    exit
exit
```

NOTES:

- **profile slaac** *slaac_profile_name*: Specifies the SLAAC profile name.
- **prefix-pool** *slaac_prefix_name*: Specifies the /64 IPv6 PD prefix pool to allocate SLAAC prefix to subscribers.
- **managed-config-flag enable**: Enables M-bit for IPv6 RA packets.
- **other-config-flag enable**: Enables O-bit for IPv6 RA packets.
- **profile subscriber** *subscriber_profile_name*: Configures subscriber profiles.
- **pppoe-profile** *pppoe_profile_name*: Specifies the PPPOE-FSOL profile name.
- **slaac-profile** *slaac_profile_name*: Specifies the SLAAC-FSOL profile name.

Step 2 Use the **show subscriber pppoe detail** command to view the PPPoE sessions with SLAAC prefix.

Example:

```
bng# show subscriber pppoe detail
Tue Oct  1  13:30:17.888 UTC+00:00
subscriber-details
{
  "subResponses": [
    {
      "state": "complete",
      "key": {
        "routerID": "asr9k-1",
        "portID": "Bundle-Ether1.1",
        "outerVlan": 100,
        "macAddr": "0011.9400.ab01",
        "pppoesessionID": 13200,
        "sublabel": "17321517",
        "upSubID": "2148432560",
       },
      "slaacInfo": {
        "prefix": "3001:ab:0:bc::",
        "prefixlength": 64,
        "poolname": "pool-ipv6",
        "fsmstate": "connected",
        "profilename": "slaac-prof",
```

```
        "otherconfig": true
      }
    }
},
```

Step 3 Use the **show subscriber session detail** command to view the Session Manager (SM) sessions with SLAAC prefix.

Example:

```
bng# show subscriber session detail
Tue Oct  1  13:37:05.476 UTC+00:00
subscriber-details
{
  "subResponses": [
    {
      "subLabel": "17321517",
      "mac": "0011.9400.ab01",
      "acct-sess-id": "01085a5d",
      "sesstype": "ppp",
      "state": "established",
      "subCreateTime": "Tue, 01 Oct 2024 04:56:05 UTC",
      "pppAuditId": 4,
      "transId": "3",
      "subsAttr": {
        "attrs": {
          "port-type": "Virtual PPPoE over VLAN(36)",
          "pppoe-session-id": "13200",
          "prefix": "3001:ab:0:bc::/64",
          "protocol-type": "ppp(2)",
        }
      },
   }
}
```