Independent Submission
Request for Comments: 8772
Category: Informational
ISSN: 2070-1721

S. Hu
China Mobile
D. Eastlake 3rd
Futurewei Technologies
F. Qin
China Mobile
T. Chua
Singapore Telecommunications
D. Huang
ZTE

May 2020

The China Mobile, Huawei, and ZTE Broadband Network Gateway (BNG) Simple Control and User Plane Separation Protocol (S-CUSP)

Abstract

A Broadband Network Gateway (BNG) in a fixed wireline access network is an Ethernet-centric IP edge router and the aggregation point for subscriber traffic. Control and User Plane Separation (CUPS) for such a BNG improves flexibility and scalability but requires various communication between the User Plane (UP) and the Control Plane (CP). China Mobile, Huawei Technologies, and ZTE have developed a simple CUPS control channel protocol to support such communication: the Simple Control and User Plane Separation Protocol (S-CUSP). S-CUSP is defined in this document.

This document is not an IETF standard and does not have IETF consensus. S-CUSP is presented here to make its specification conveniently available to the Internet community to enable diagnosis and interoperability.

Status of This Memo

This document is not an Internet Standards Track specification; it is published for informational purposes.

This is a contribution to the RFC Series, independently of any other RFC stream. The RFC Editor has chosen to publish this document at its discretion and makes no statement about its value for implementation or deployment. Documents approved for publication by the RFC Editor are not candidates for any level of Internet Standard; see Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc8772.

Table of Contents

1. Introduction

A Broadband Network Gateway (BNG) in a fixed wireline access network is an Ethernet-centric IP edge router and the aggregation point for subscriber traffic. To provide centralized session management, flexible address allocation, high scalability for subscriber management capacity, and cost-efficient redundancy, the CU-separated (CP/UP-separated) BNG framework is described in a technical report [TR-384] from the Broadband Forum (BBF). The CU-separated service CP, which is responsible for user access authentication and setting forwarding entries in UPs, can be virtualized and centralized. The routing control and forwarding plane, i.e., the BNG UP (local), can be distributed across the infrastructure. Other structures can also be supported, such as the CP and UP being virtual or both being physical.

Note: In this document, the terms "user" and "subscriber" are used interchangeably.

This document specifies the Simple CU Separation Protocol (S-CUSP) for communications over the BNG control channel between a BNG CP and a set of UPs. S-CUSP is designed to be flexible and extensible so as to allow for easy addition of messages and data items, should further requirements be expressed in the future.

This document is not an IETF standard and does not have IETF consensus. S-CUSP was designed by China Mobile, Huawei Technologies, and ZTE. It is presented here to make the S-CUSP specification conveniently available to the Internet community to enable diagnosis and interoperability.

At the time of writing this document, the BBF is working to produce [WT-459], which will describe an architecture and requirements for a CP and UP separation of a disaggregated BNG. Future work may attempt to show how the protocol described in this document addresses those requirements and may modify this specification to handle unaddressed requirements.

2. Terminology

This section specifies implementation requirement keywords and terms used in this document. S-CUSP messages are described in this document using Routing Backus-Naur Form (RBNF) as defined in [RFC5511].

2.1. Implementation Requirement Keywords

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2.2. Terms

This section specifies terms used in this document.

AAA: Authentication Authorization Accounting.

ACK: Acknowledgement message.

BAS: Broadband Access Server, also known as a BBRAS, BNG, or BRAS.

BNG: Broadband Network Gateway. A BNG (or Broadband Remote Access Server (BRAS)) routes traffic to and from broadband remote access devices such as digital subscriber line access multiplexers (DSLAM) on an Internet Service Provider's (ISP) network. BNG / BRAS can also be referred to as a BAS or BBRAS.

BRAS: Broadband Remote Access Server, also known as a BAS, BBRAS, or BNG.

CAR: Committed Access Rate.

CBS: Committed Burst Size.

CGN: Carrier Grade NAT.

Ci: Control Interface.

CIR: Committed Information Rate.

CoA: Change of Authorization.

CP: Control Plane. CP is a user control management component that supports the management of the UP's resources such as the user entry and forwarding policy.

CU: Control Plane / User Plane.

CUSP: Control and User Plane Separation Protocol.

DEI: Drop Eligibility Indicator as defined in [802.1Q]. A bit in a VLAN tag after the priority and before the VLAN ID. (This bit was formerly the CFI (Canonical Format Indicator).)

DHCP: Dynamic Host Configuration Protocol [RFC2131].

dial-up: This refers to the initial connection messages when a new subscriber appears. The name is left over from when subscribers literally dialed up on a modem-equipped phone line but herein is applied to other initial connection techniques. Initial connection is frequently indicated by the receipt of packets over PPPoE [RFC2516] or IPoE.

EMS: Element Management System.

IPoE: IP over Ethernet.

L2TP: Layer 2 Tunneling Protocol [RFC2661].

LAC: L2TP Access Concentrator.

LNS: L2TP Network Server.

MAC: 48-bit Media Access Control address [RFC7042].

MANO: Management and Orchestration.

Mi: Management Interface.

MSS: Maximum Segment Size.

MRU: Maximum Receive Unit.

NAT: Network Address Translation [RFC3022].

ND: Neighbor Discovery.

NFV: Network Function Virtualization.

NFVI: NFV Infrastructure.

PBS: Peak Burst Size.

PD: Prefix Delegation.

PIR: Peak Information Rate.

PPP: Point-to-Point Protocol [RFC1661].

PPPoE: PPP over Ethernet [RFC2516].

RBNF: Routing Backus-Naur Form [RFC5511].

RG: Residential Gateway.

S-CUSP: Simple Control and User Plane Separation Protocol.

Subscriber: The remote user gaining network accesses via a BNG.

Si: Service Interface.

TLV: Type-Length-Value. See Sections 7.1 and 7.3.

UP: User Plane. UP is a network edge and user policy implementation component. The traditional router's control plane and forwarding plane are both preserved on BNG devices in the form of a user plane.

URPF: Unicast Reverse Path Forwarding.

User: Equivalent to "customer" or "subscriber".

VRF: Virtual Routing and Forwarding.

3. BNG CUPS Overview

3.1. BNG CUPS Motivation

The rapid development of new services, such as 4K TV, Internet of Things (IoT), etc., and increasing numbers of home broadband service users present some new challenges for BNGs such as:

Low resource utilization: The traditional BNG acts as both a gateway for user access authentication and accounting and also an IP network's Layer 3 edge. The mutually affecting nature of the tightly coupled control plane and forwarding plane makes it difficult to achieve the maximum performance of either plane.

Complex management and maintenance: Due to the large numbers of traditional BNGs, configuring each device in a network is very tedious when deploying global service policies. As the network expands and new services are introduced, this deployment mode will cease to be feasible as it is unable to manage services effectively and to rectify faults rapidly.

Slow service provisioning: The coupling of the CP and the forwarding plane, in addition to being a distributed network control

mechanism, means that any new technology has to rely heavily on the existing network devices.

The framework for a cloud-based BNG with CU separation to address these challenges for fixed networks is described in [TR-384]. The main idea of CU separation is to extract and centralize the user management functions of multiple BNG devices, forming a unified and centralized CP. The traditional router's CP and forwarding plane are both preserved on BNG devices in the form of a UP.

3.2. BNG CUPS Architecture Overview

The functions in a traditional BNG can be divided into two parts: (1) the user access management function and (2) the routing function. The user access management function can be deployed as a centralized module or device, called the BNG Control Plane (BNG-CP). The routing function, which includes routing control and the forwarding engine, can be deployed in the form of the BNG User Plane (BNG-UP).

Figure 1 shows the architecture of a CU-separated BNG:

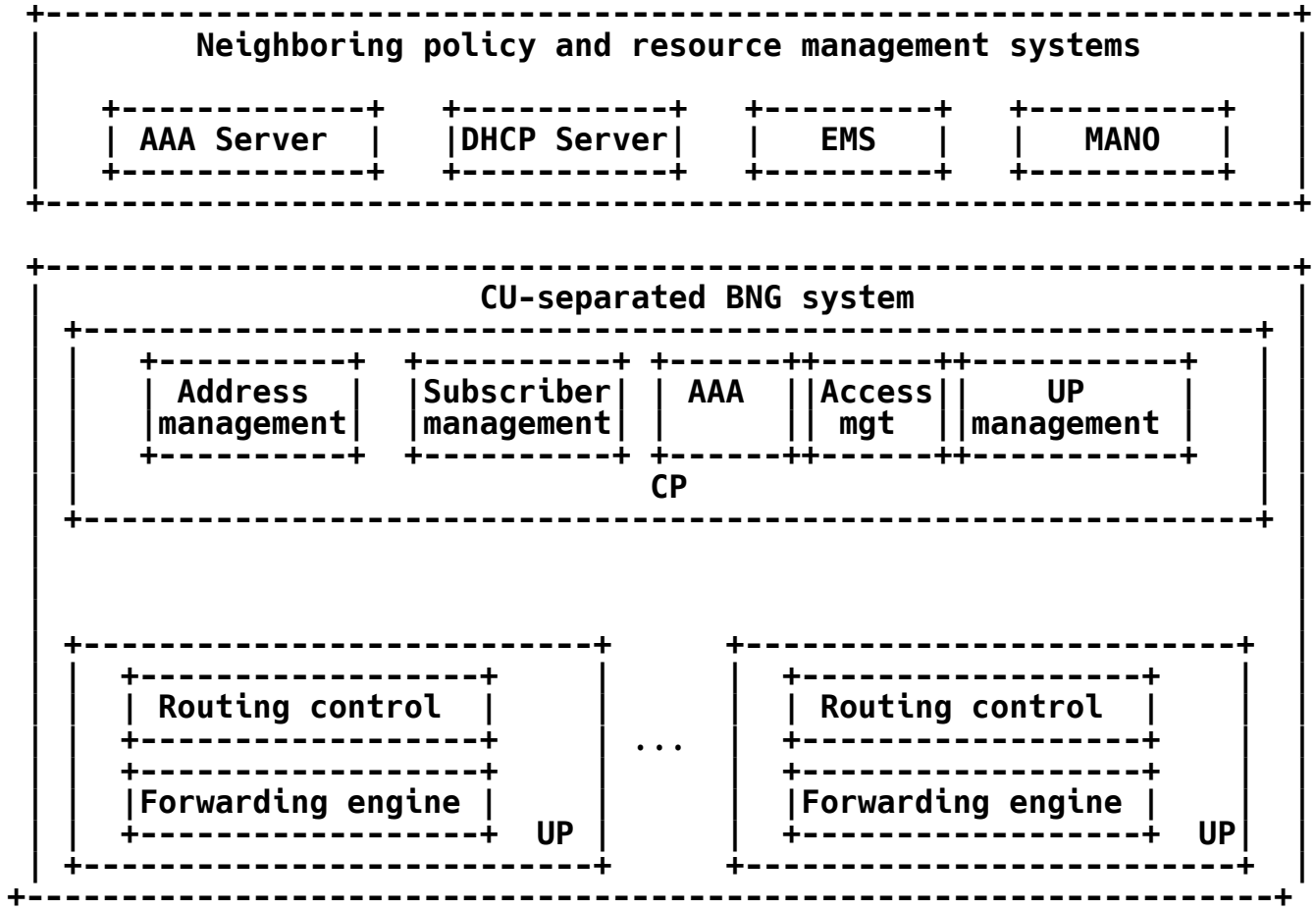


Figure 1: Architecture of a CU-Separated BNG

As shown in Figure 1, the BNG-CP could be virtualized and centralized, which provides benefits such as centralized session management, flexible address allocation, high scalability for

subscriber management capacity, cost-efficient redundancy, etc. The functional components inside the BNG-CP can be implemented as Virtual Network Functions (VNFs) and hosted in an NFVI.

The UP management module in the BNG-CP centrally manages the distributed BNG-UPs (e.g., load balancing), as well as the setup, deletion, and maintenance of channels between CPs and UPs. Other modules in the BNG-CP, such as address management, AAA, etc., are responsible for the connection with external subsystems in order to fulfill those services. Note that the UP SHOULD support both physical and virtual network functions. For example, network functions related to BNG-UP L3 forwarding can be disaggregated and distributed across the physical infrastructure, and the other CP management functions in the CU-separated BNG can be moved into the NFVI for virtualization [TR-384].

The details of the CU-separated BNG's function components are as follows:

The CP is responsible for the following:

* Address management: Unified address pool management and CGN subscriber address traceability management.

* AAA: This component performs Authentication, Authorization, and Accounting, together with RADIUS/Diameter. The BNG communicates with the AAA server to check whether the subscriber who sent an access request has network access authority. Once the subscriber goes online, this component (together with the Service Control component) implements accounting, data capacity limitation, and QoS enforcement policies.

* Subscriber management: User entry management and forwarding policy management.

* Access management: Process user dial-up packets, such as PPPoE, DHCP, L2TP, etc.

* UP management: Management of UP interface status and the setup, deletion, and maintenance of channels between CP and UP.

The UP is responsible for the following:

* Routing control functions: Responsible for instantiating routing forwarding plane (e.g., routing, multicast, MPLS, etc.).

* Routing and service forwarding plane functions: Responsibilities include traffic forwarding, QoS, and traffic statistics collection.

* Subscriber detection: Responsible for detecting whether a subscriber is still online.

3.3. BNG CUPS Interfaces

The three interfaces defined below support the communication between

the CP and UP. These are referred to as the Service Interface (Si), Control Interface (Ci), and Management Interface (Mi) as shown in Figure 2.

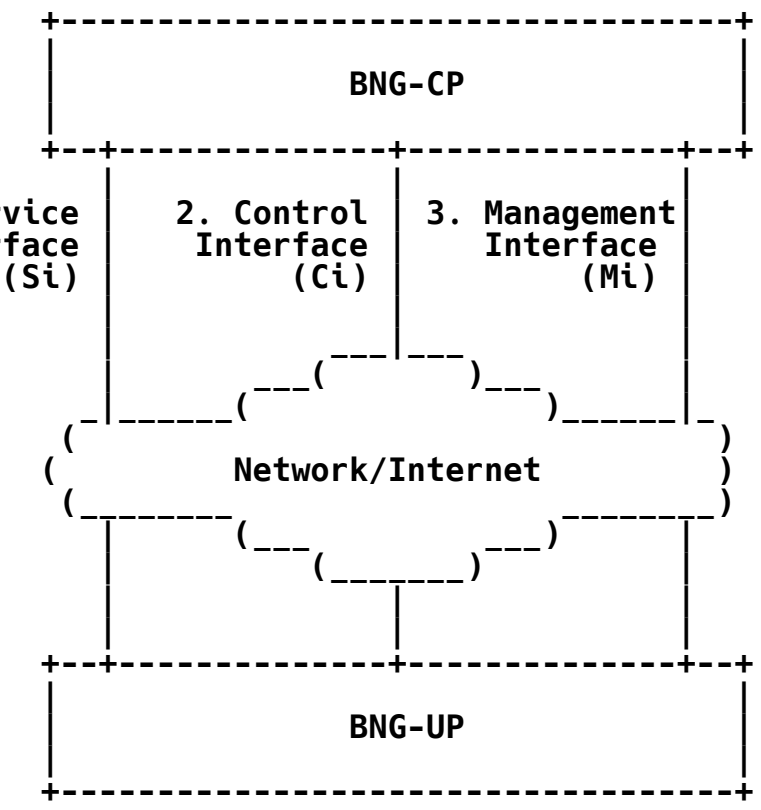


Figure 2: Interfaces between the CP and UP of the BNG

3.3.1. Service Interface (Si)

For a traditional BNG (without CU separation), the user dial-up signals are terminated and processed by the CP of a BNG. When the CP and UP of a BNG are separated, there needs to be a way to relay these signals between the CP and the UP.

The Si is used to establish tunnels between the CP and UP. The tunnels are responsible for relaying the PPPoE-, IPoE-, and L2TP-related control packets that are received from a Residential Gateway (RG) over those tunnels. An appropriate tunnel type is Virtual eXtensible Local Area Network (VXLAN) [RFC7348].

The detailed definition of Si is out of scope for this document.

3.3.2. Control Interface (Ci)

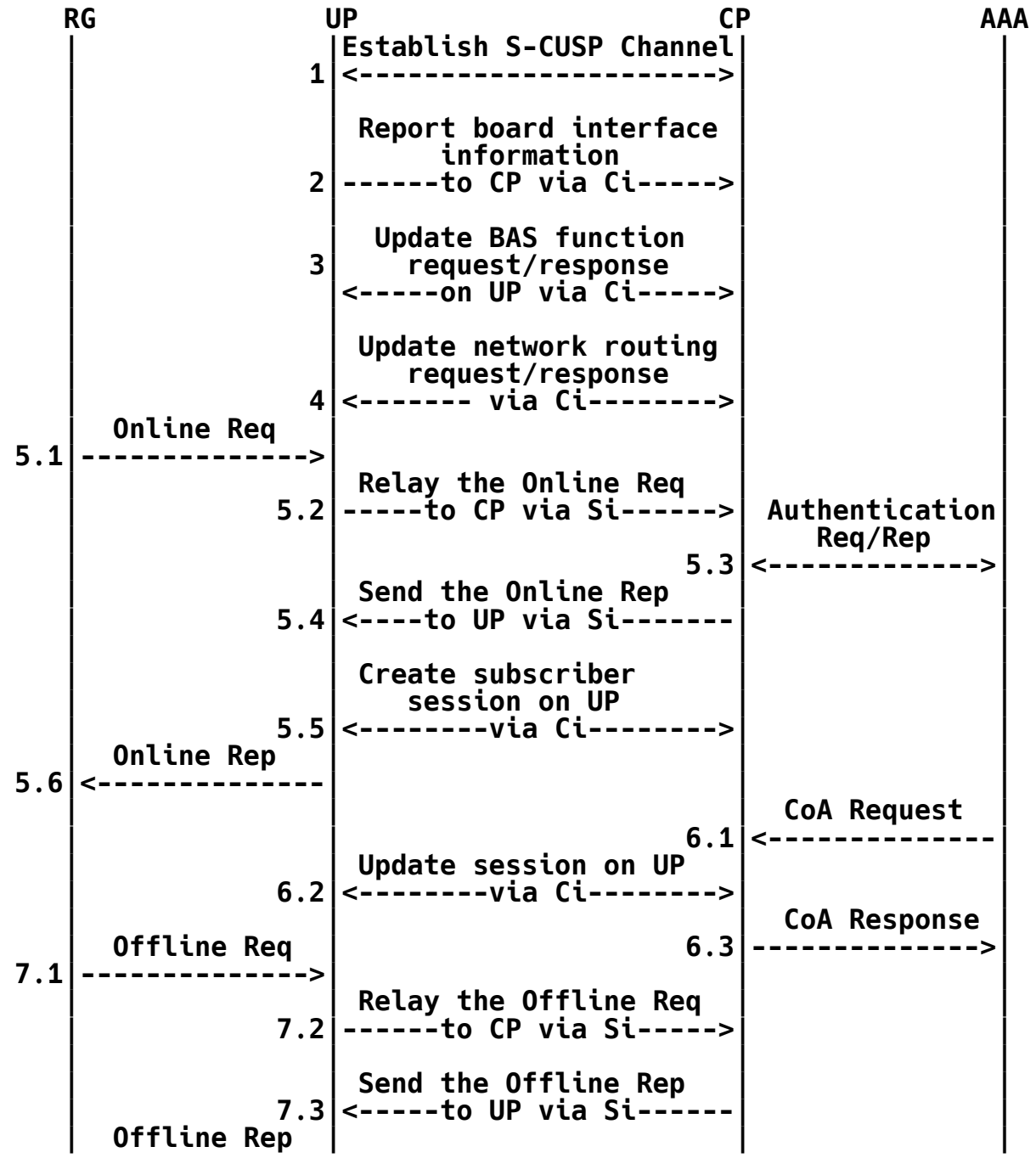
The CP uses the Ci to deliver subscriber session states, network routing entries, etc., to the UP (see Section 6.2.7). The UP uses this interface to report subscriber service statistics, subscriber detection results, etc., to the CP (see Sections 6.3 and 6.4). A carrying protocol for this interface is specified in this document.

3.3.3. Management Interface (Mi)

The Network Configuration Protocol (NETCONF) [RFC6241] is the protocol used on the Mi between a CP and UP. It is used to configure the parameters of the Ci, Si, access interfaces, and QoS/ACL Templates. It is expected that implementations will make use of existing YANG models where possible but that new YANG models specific to S-CUSP will need to be defined. The definitions of the parameters that can be configured are out of scope for this document.

3.4. BNG CUPS Procedure Overview

The following numbered sequences (Figure 3) give a high-level view of the main BNG CUPS procedures.



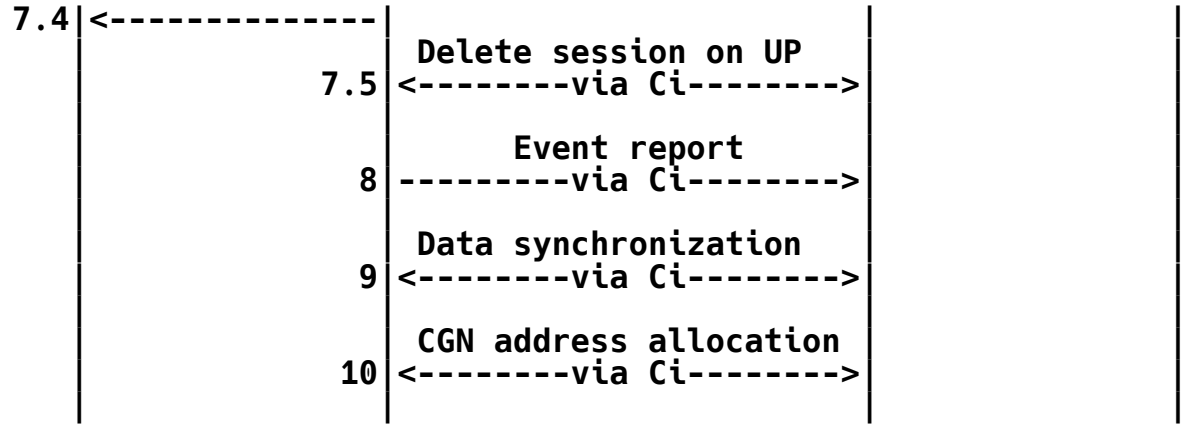


Figure 3: BNG CUPS Procedures Overview

(1) S-CUSP session establishment: This is the first step of the BNG CUPS procedures. Once the Ci parameters are configured on a UP, it will start to set up S-CUSP sessions with the specified CPs. The detailed definition of S-CUSP session establishment can be found in Section 4.1.1.

(2) Board and interface report: Once the S-CUSP session is established between the UP and a CP, the UP will report status information on the boards and subscriber-facing interfaces of this UP to the CP. A board can also be called a Line/Service Process Unit (LPU/SPU) card. The subscriber-facing interfaces refer to the interfaces that connect the access network nodes (e.g., Optical Line Terminal (OLT), DSLAM, etc.). The CP can use this information to enable the Broadband Access Server (BAS) function (e.g., IPoE, PPPoE, etc.) on the specified interfaces. See Sections 4.2.1 and 7.10 for more details on resource reporting.

(3) BAS function enable: To enable the BAS function on the specified interfaces of a UP.

(4) Subscriber network route advertisement: The CP will allocate one or more IP address blocks to a UP. Each address block contains a series of IP addresses. Those IP addresses will be allocated to subscribers who are dialing up from the UP. To enable other nodes in the network to learn how to reach the subscribers, the CP needs to notify the UP to advertise to the network the routes that can reach those IP addresses.

(5) 5.1-5.6 is a complete call flow of a subscriber dial-up (as defined in Section 4.3.1) process. When a UP receives a dial-up request, it will relay the request packet to a CP through the Si. The CP will parse the request. If everything is OK, it will send an authentication request to the AAA server to authenticate the subscriber. Once the subscriber passes the authentication, the AAA server will return a positive response to the CP. Then the CP will send the dial-up response packet to the UP, and the UP will forward the response packet to the subscriber (RG). At the same time, the CP will create a subscriber session on the UP, enabling the subscriber to access

the network. For different access types, the process may be a bit different, but the high-level process is similar. For each access type, the detailed process can be found in Section 5.

(6) 6.1-6.3 is the sequence when updating an existing subscriber session. The AAA server initiates a Change of Authorization (CoA) and sends the CoA to the CP. The CP will then update the session according to the CoA. See Section 4.3.2 for more detail on CP messages updating UP tables.

(7) 7.1-7.5 is the sequence for deleting an existing subscriber session. When a UP receives an Offline Request, it will relay the request to a CP through the Si. The CP will send back a response to the UP through the Si. The UP will then forward the Offline Response to the subscriber. Then the CP will delete the session on the UP through the Ci.

(8) Event reports include the following two parts (more detail can be found in Section 4.3.4). Both are reported using the Event message:

8.1. Subscriber Traffic Statistics Report

8.2. Subscriber Detection Result Report

(9) Data synchronization: See Section 4.2.5 for more detail on CP and UP synchronization.

(10) CGN address allocation: See Section 4.2.4 for more detail on CGN address allocation.

4. S-CUSP Protocol Overview

4.1. Control Channel Procedures

4.1.1. S-CUSP Session Establishment

A UP is associated with a CP and is controlled by that CP. In the case of a hot-standby or cold-standby, a UP is associated with two CPs: the master CP and standby CP. The association between a UP and its CPs is implemented by dynamic configuration.

Once a UP knows its CPs, the UP starts to establish S-CUSP sessions with those CPs, as shown in Figure 4.

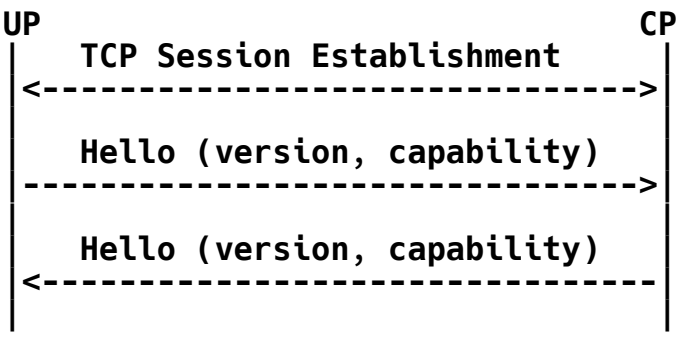


Figure 4: S-CUSP Session Establishment

The S-CUSP session establishment consists of two successive steps:

(1) Establishment of a TCP connection (3-way handshake) [RFC793] between the CP and the UP using a configured port from the dynamic port range (49152-65535).

(2) Establishment of an S-CUSP session over the TCP connection.

Once the TCP connection is established, the CP and the UP initialize the S-CUSP session, during which the version and Keepalive timers are negotiated.

The version information (Hello TLV, see Section 7.4) is carried within Hello messages (see Section 6.2.1). A CP can support multiple versions, but a UP can only support one version; thus the version negotiation is based on whether a version can be supported by both the CP and the UP. If a CP or UP receives a Hello message that does not indicate a version supported by both, it responds with a Hello message containing an Error Information TLV to notify the peer of the Version-Mismatch error, and the session establishment phase fails.

Keepalive negotiation is performed by carrying a Keepalive TLV in the Hello message. The Keepalive TLV includes a Keepalive timer and DeadTimer field. The CP and UP have to agree on the Keepalive Timer and DeadTimer. Otherwise, a subsequent Hello message with an Error Information TLV will be sent to its peer, and the session establishment phase fails.

The S-CUSP session establishment phase fails if the CP or UP disagree on the version and keepalive parameters or if one of the CP or UP does not answer after the expiration of the Establishment timer. When the S-CUSP session establishment fails, the TCP connection is promptly closed. Successive retries are permitted, but an implementation SHOULD make use of an exponential backoff session establishment retry procedure.

The S-CUSP session timer values that need to be configured are summarized in Table 1.

Timer Name	Range in Seconds	Default Value
Establishment Timer	1-32767	45
Keepalive Timer	0-255	30
DeadTimer	1-32767	4 * Keepalive

Table 1: S-CUSP Session Timers

4.1.2. Keepalive Timer and DeadTimer

Once an S-CUSP session has been established, a UP or CP may want to

know that its S-CUSP peer is still connected.

Each end of an S-CUSP session runs a Keepalive timer. It restarts the timer every time it sends a message on the session. When the timer expires, it sends a Keepalive message. Thus, a message is transmitted at least as often as the value to which the Keepalive timer is reset, unless, as explained below, that value is the special value zero.

Each end of an S-CUSP session also runs a DeadTimer and restarts that DeadTimer whenever a message is received on the session. If the DeadTimer expires at an end of the session, that end declares the session dead and the session will be closed, unless their DeadTimer is set to the special value zero, in which case the session will not time out.

The minimum value of the Keepalive timer is 1 second, and it is specified in units of 1 second. The RECOMMENDED default value is 30 seconds. The recommended default for the DeadTimer is four times the value of the Keepalive timer used by the remote peer. As above, the timers may be disabled by setting them to zero.

The Keepalive timer and DeadTimer are negotiated through the Keepalive TLV carried in the Hello message.

4.2. Node Procedures

4.2.1. UP Resource Report

Once an S-CUSP session has been established between a CP and a UP, the UP reports the state information of the boards and access-facing interfaces on the UP to the CP, as shown in Figure 5. Report messages are unacknowledged and are assumed to be delivered because the session runs over TCP.

The CP can use that information to activate/enable the BAS functions (e.g., IPoE, PPPoE, etc.) on the specified interfaces.

In addition, the UP resource report may trigger a UP warm-standby process. In the case of warm-standby, a failure on a UP may trigger the CP to start a warm-standby process, by moving the online subscriber sessions to a standby UP and then directing the affected subscribers to access the Internet through the standby UP.

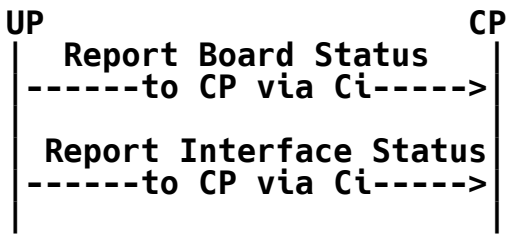


Figure 5: UP Board and Interface Report

Board status information is carried in the Board Status TLV (Section 7.10.2), and interface status information is carried in the

Interface Status TLV (Section 7.10.1). Both Board Status and Interface Status TLVs are carried in the Report message (Section 6.4).

4.2.2. Update BAS Function on Access Interface

Once the CP collects the interface status of a UP, it will activate/deactivate/modify the BAS functions on specified interfaces through the Update_Request and Update_Response message exchanges (Section 6.2), carrying the BAS Function TLV (Section 7.7).

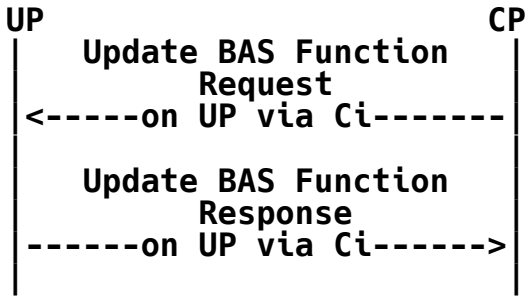


Figure 6: Update BAS Function

4.2.3. Update Network Routing

The CP will allocate one or more address blocks to a UP. Each address block contains a series of IP addresses. Those IP addresses will be assigned to subscribers who are dialing up to the UP. To enable the other nodes in the network to learn how to reach the subscribers, the CP needs to install the routes on the UP and notify the UP to advertise the routes to the network.

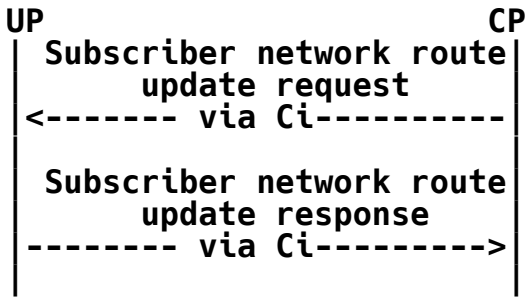


Figure 7: Update Network Routing

The Update_Request and Update_Response message exchanges, carrying the IPv4/IPv6 Routing TLVs (Section 7.8), update the subscriber's network routing information.

4.2.4. CGN Public IP Address Allocation

The following sequences (Figure 8) describe the procedures related to CGN address management. Three independent procedures are defined: one each for CGN address allocation request/response, CGN address renewal request/response, and CGN address release request/response.

CGN address allocation/renew/release procedures are designed for the

case where the CGN function is running on the UP. The UP has to map the subscriber private IP addresses to public IP addresses, and such mapping is performed by the UP locally when a subscriber dials up. That means the UP has to ask for public IPv4 address blocks for CGN subscribers from the CP.

In addition, when a public IP address is allocated to a UP, there will be a lease time (e.g., one day). Before the lease time expires, the UP can ask for renewal of the IP address lease from the CP. It is achieved by the exchange of the Addr_Renew_Req and Addr_Renew_Ack messages.

If the public IP address will not be used anymore, the UP SHOULD release the address by sending an Addr_Release_Req message to the CP.

If the CP wishes to withdraw addresses that it has previously leased to a UP, it uses the same procedures as above. The Oper code (see Section 7.1) in the IPv4/IPv6 Routing TLV (see Section 7.8) determines whether the request is an update or withdraw.

The relevant messages are defined in Section 6.5.

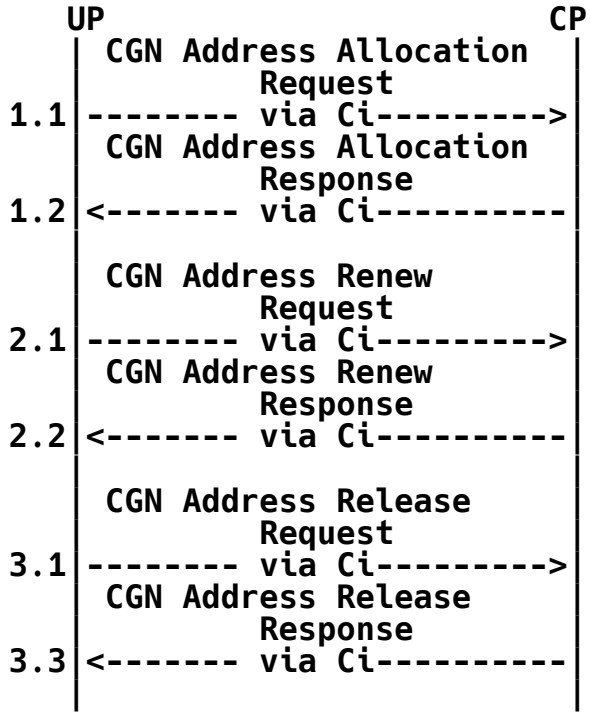


Figure 8: CGN Public IP Address Allocation

4.2.5. Data Synchronization between the CP and UP

For a CU-separated BNG, the UP will continue to function using the state that has been installed in it even if the CP fails or the session between the UP and CP fails.

Under some circumstances, it is necessary to synchronize state between the CP and UP, for example, if a CP fails and the UP is switched to a different CP.

Synchronization includes two directions. One direction is from UP to CP; in that case, the synchronization information is mainly about the board/interface status of the UP. The other direction is from CP to UP; in that case, the subscriber sessions, subscriber network routes, L2TP tunnels, etc., will be synchronized to the UP.

The synchronization is triggered by a Sync_Request message, to which the receiver will (1) reply with a Sync_Begin message to notify the requester that synchronization will begin and (2) then start the synchronization using the Sync_Data message. When synchronization finishes, a Sync_End message will be sent.

Figure 9 shows the process of data synchronization between a UP and a CP.

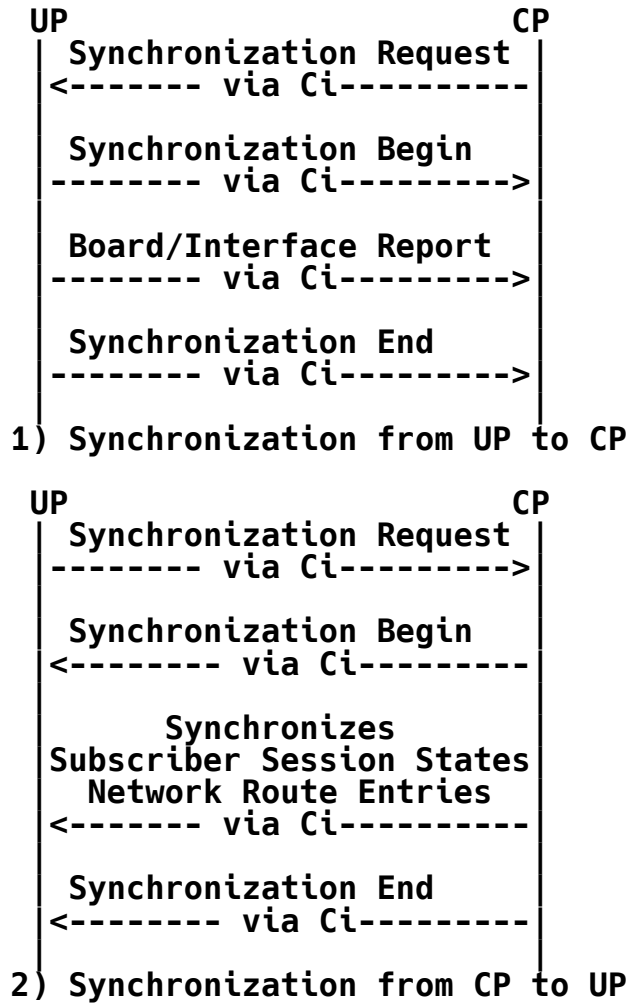


Figure 9: Data Synchronization

4.3. Subscriber Session Procedures

A subscriber session consists of a set of forwarding states, policies, and security rules that are applied to the subscriber. It is used for forwarding subscriber traffic in a UP. To initialize a session on a UP, a collection of hardware resources (e.g., NP, TCAM,

etc.) has to be allocated to a session on a UP as part of its initiation.

Procedures related to subscriber sessions include subscriber session creation, update, deletion, and statistics reporting. The following subsections give a high-level view of the procedures.

4.3.1. Create Subscriber Session

The sequence below (Figure 10) describes the DHCP IPv4 dial-up process. It is an example that shows how a subscriber session is created. (An example for IPv6 appears in Section 5.1.2.)

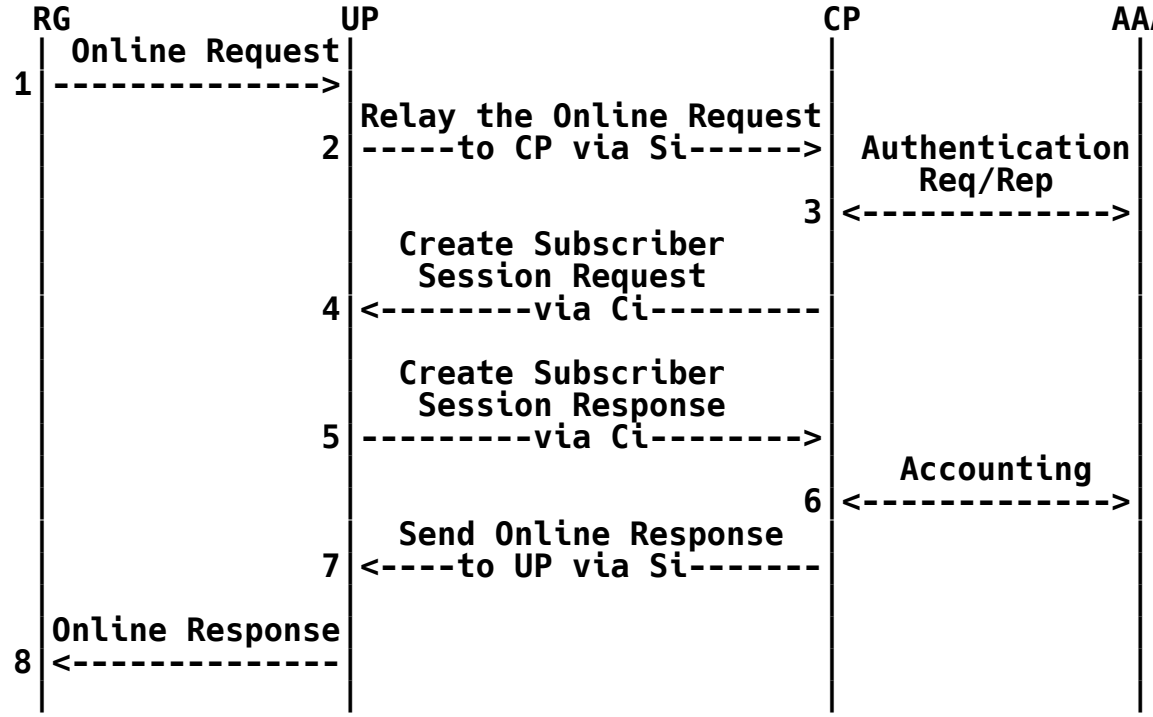


Figure 10: Creating a Subscriber Session

The request starts from an Online Request message (step 1) from the RG (for example, a DHCP Discovery packet). When the UP receives the Online Request from the RG, it will tunnel the Online Request to the CP through the Si (step 2). A tunneling technology implements the Si.

When the CP receives the Online Request from the UP, it will send an authentication request to the AAA server to authenticate and authorize the subscriber (step 3). When a positive reply is received from the AAA server, the CP starts to create a subscriber session for the request. Relevant resources (e.g., IP address, bandwidth, etc.) will be allocated to the subscriber. Policies and security rules will be generated for the subscriber. Then the CP sends a request to create a session to the UP through the Ci (step 4), and a response is expected from the UP to confirm the creation (step 5).

Finally, the CP will notify the AAA server to start accounting (step 6). At the same time, an Online Response message (for example, a

DHCP Ack packet) will be sent to the UP through the Si (step 7). The UP will then forward the Online Response to the RG (step 8).

That completes the subscriber activation process.

4.3.2. Update Subscriber Session

The following numbered sequence (Figure 11) shows the process of updating the subscriber session.

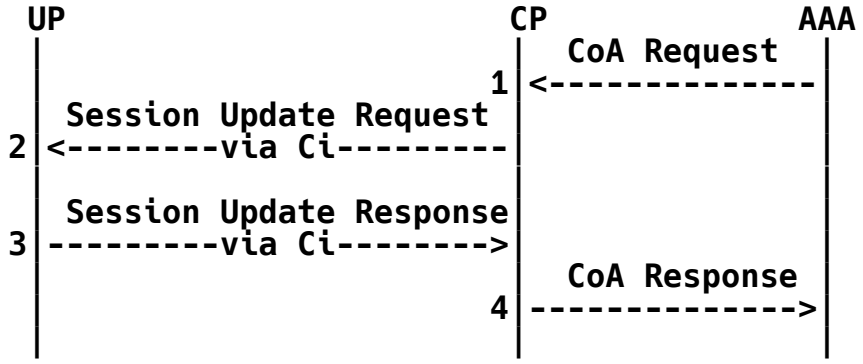


Figure 11: Updating a Subscriber Session

When a subscriber session has been created on a UP, there may be requirements to update the session with new parameters (e.g., bandwidth, QoS, policies, etc.).

This procedure is triggered by a Change of Authorization (CoA) request message sent by the AAA server. The CP will update the session on the UP according to the new parameters through the Ci.

4.3.3. Delete Subscriber Session

The call flow below shows how S-CUSP deals with a subscriber Offline Request.

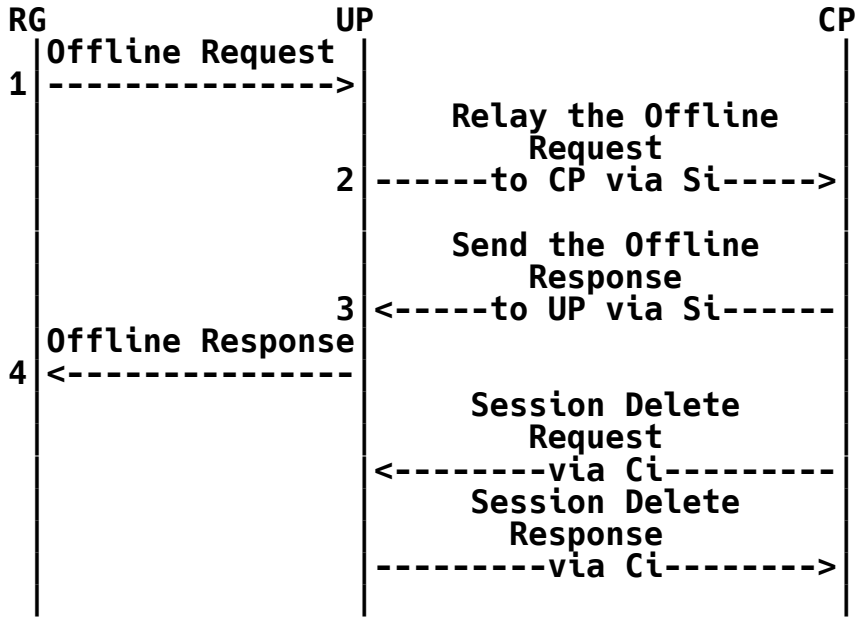


Figure 12: Deleting a Subscriber Session

Similar to the session creation process, when a UP receives an Offline Request from an RG, it will tunnel the request to a CP through the Si.

When the CP receives the Offline Request, it will withdraw/release the resources (e.g., IP address, bandwidth) that have been allocated to the subscriber. It then sends a reply to the UP through the Si, and the UP will forward the reply to the RG. At the same time, it will delete all the status of the session on the UP through the Ci.

4.3.4. Subscriber Session Events Report

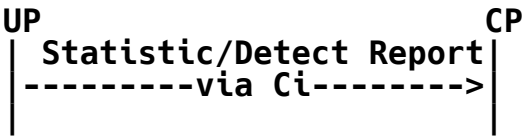


Figure 13: Events Report

When a session is created on a UP, the UP will periodically report statistics information and subscriber detection results of the session to the CP.

5. S-CUSP Call Flows

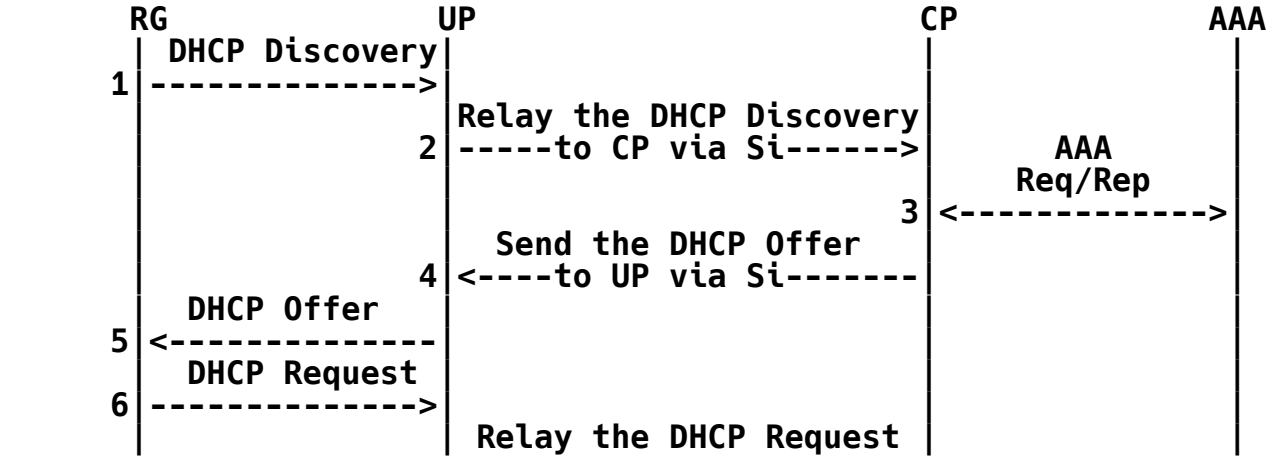
The subsections below give an overview of various "dial-up" interactions over the Si followed by an overview of the setting of information in the UP by the CP using S-CUSP over the Ci.

S-CUSP messages are described in this document using Routing Backus Naur Form (RBNF) as defined in [RFC5511].

5.1. IPoE

5.1.1. DHCPv4 Access

The following sequence (Figure 14) shows detailed procedures for DHCPv4 access.



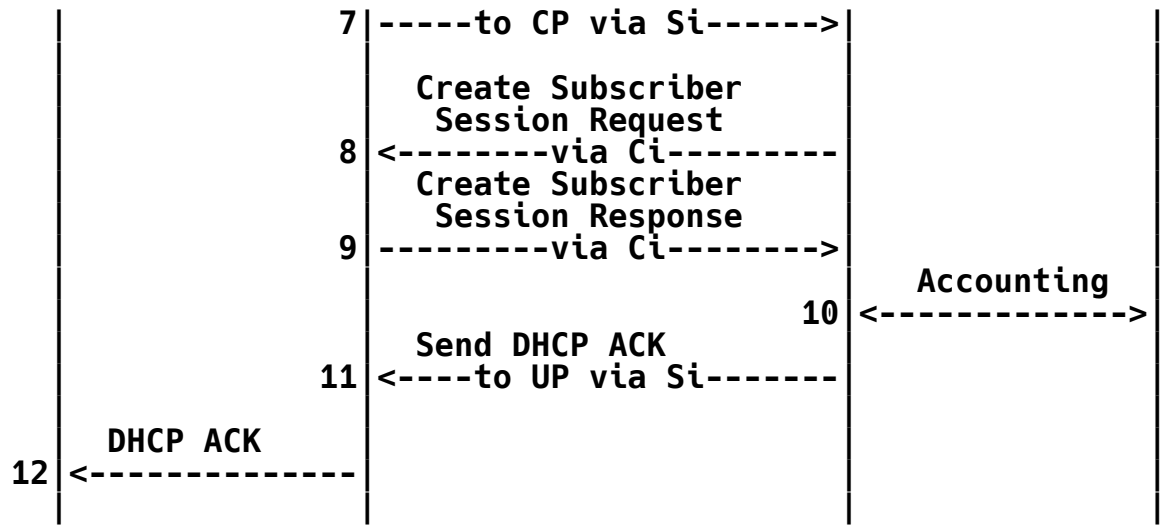


Figure 14: DHCPv4 Access

S-CUSP implements steps 8 and 9.

After a subscriber is authenticated and authorized by the AAA server, the CP creates a new subscriber session on the UP. This is achieved by sending an Update_Request message to the UP.

The format of the Update_Request message is shown as follows using RBNF:

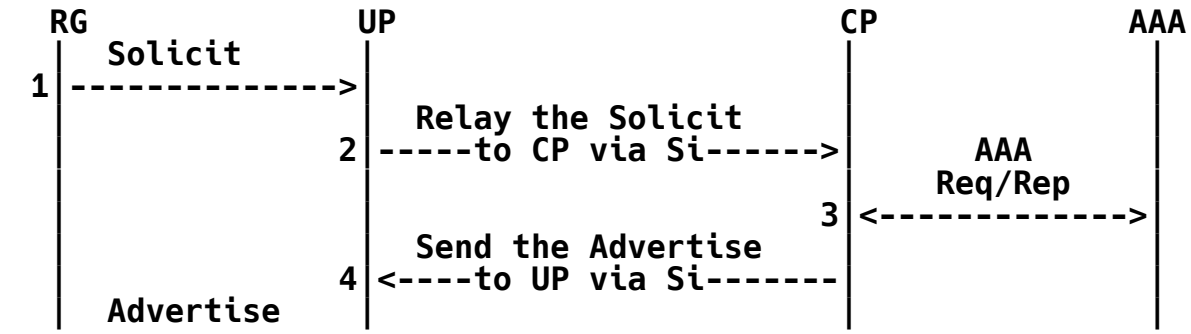
```
<Update_Request Message> ::= <Common Header>
                  <Basic Subscriber TLV>
                  <IPv4 Subscriber TLV>
                  <IPv4 Routing TLV>
                  [<Subscriber Policy TLV>]
```

The UP will reply with an Update_Response message. The format of the Update_Response message is as follows:

```
<Update_Response Message> ::= <Common Header>
                 <Update Response TLV>
                 [<Subscriber CGN Port Range TLV>]
```

5.1.2. DHCPv6 Access

The following sequence (Figure 15) shows detailed procedures for DHCPv6 access.



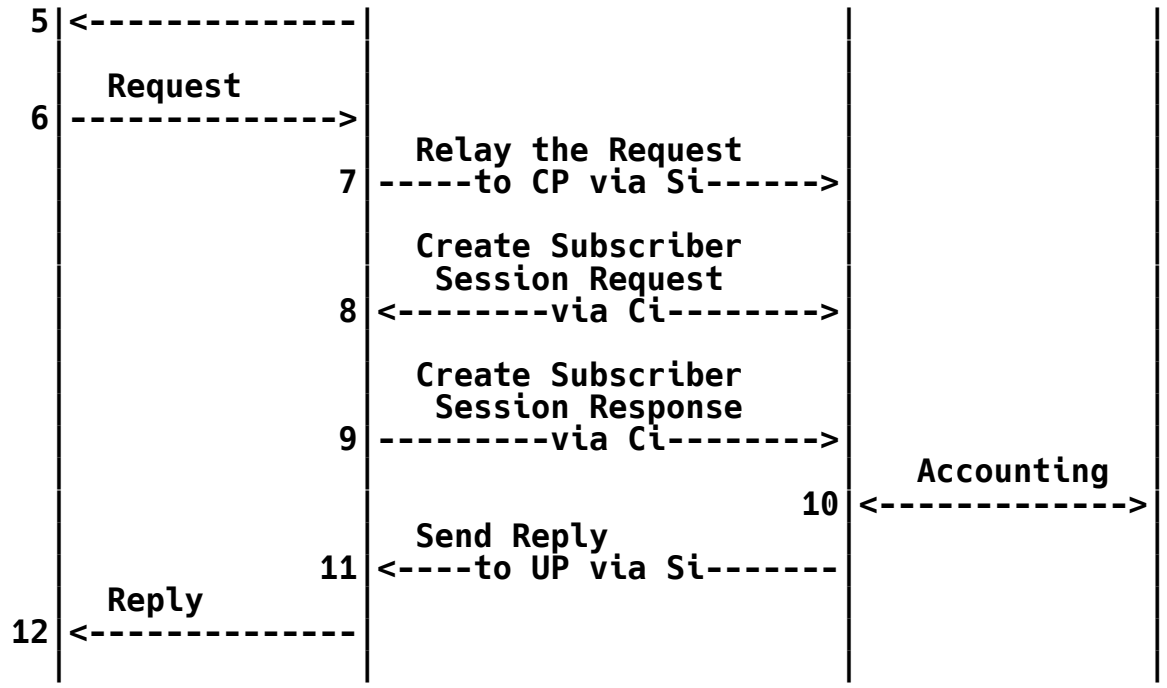


Figure 15: DHCPv6 Access

Steps 1-7 are a standard DHCP IPv6 access process. The subscriber creation is triggered by a DHCP IPv6 request message. When this message is received, it means that the subscriber has passed the AAA authentication and authorization. Then the CP will create a subscriber session on the UP. This is achieved by sending an Update_Request message to the UP (step 8).

The format of the Update_Request message is as follows:

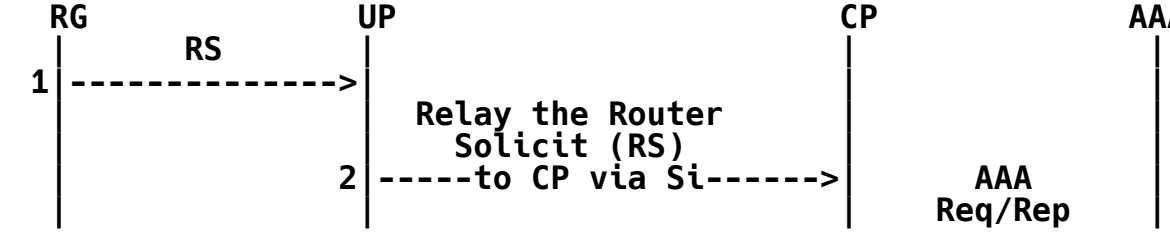
```
<Update_Request Message> ::= <Common Header>
                  <Basic Subscriber TLV>
                  <IPv6 Subscriber TLV>
                  <IPv6 Routing TLV>
                  [<Subscriber Policy TLV>]
```

The UP will reply with an Update_Response message (step 9). The format of the Update_Response message is as follows:

```
<Update_Response Message> ::= <Common Header>
                 <Update Response TLV>
```

5.1.3. IPv6 Stateless Address Autoconfiguration (SLAAC) Access

The following flow (Figure 16) shows the IPv6 SLAAC access process.



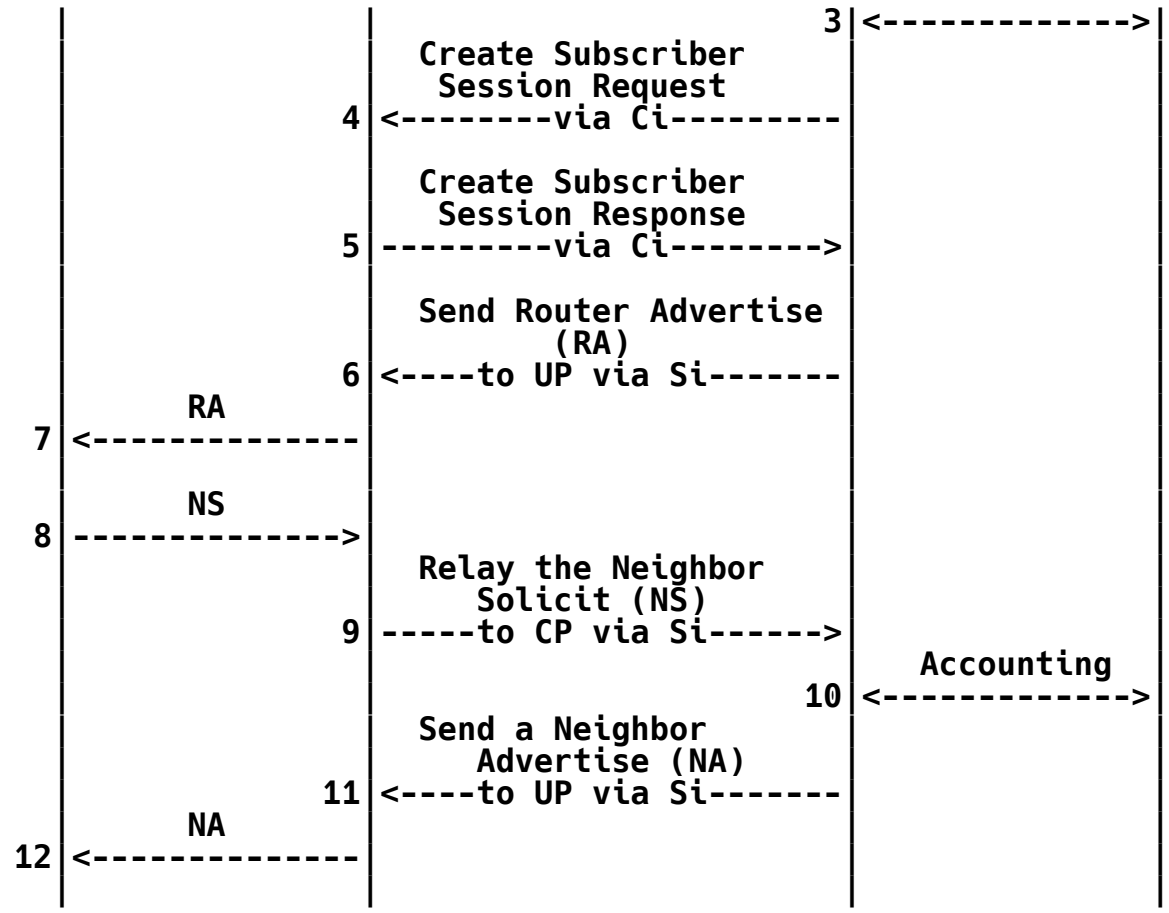


Figure 16: IPv6 SLAAC Access

It starts with a Router Solicit (RS) request from an RG that is tunneled to the CP by the UP. After the AAA authentication and authorization, the CP will create a subscriber session on the UP.

This is achieved by sending an Update_Request message to the UP (step 4).

The format of the Update_Request message is as follows:

```
<Update_Request Message> ::= <Common Header>
                  <Basic Subscriber TLV>
                  <IPv6 Subscriber TLV>
                  <IPv6 Routing TLV>
                  [<Subscriber Policy TLV>]
```

The UP will reply with an Update_Response message (step 5). The format of the Update_Response message is as follows:

```
<Update_Response Message> ::= <Common Header>
                 <Update Response TLV>
```

5.1.4. DHCPv6 and SLAAC Access

The following call flow (Figure 17) shows the DHCP IPv6 and SLAAC access process.

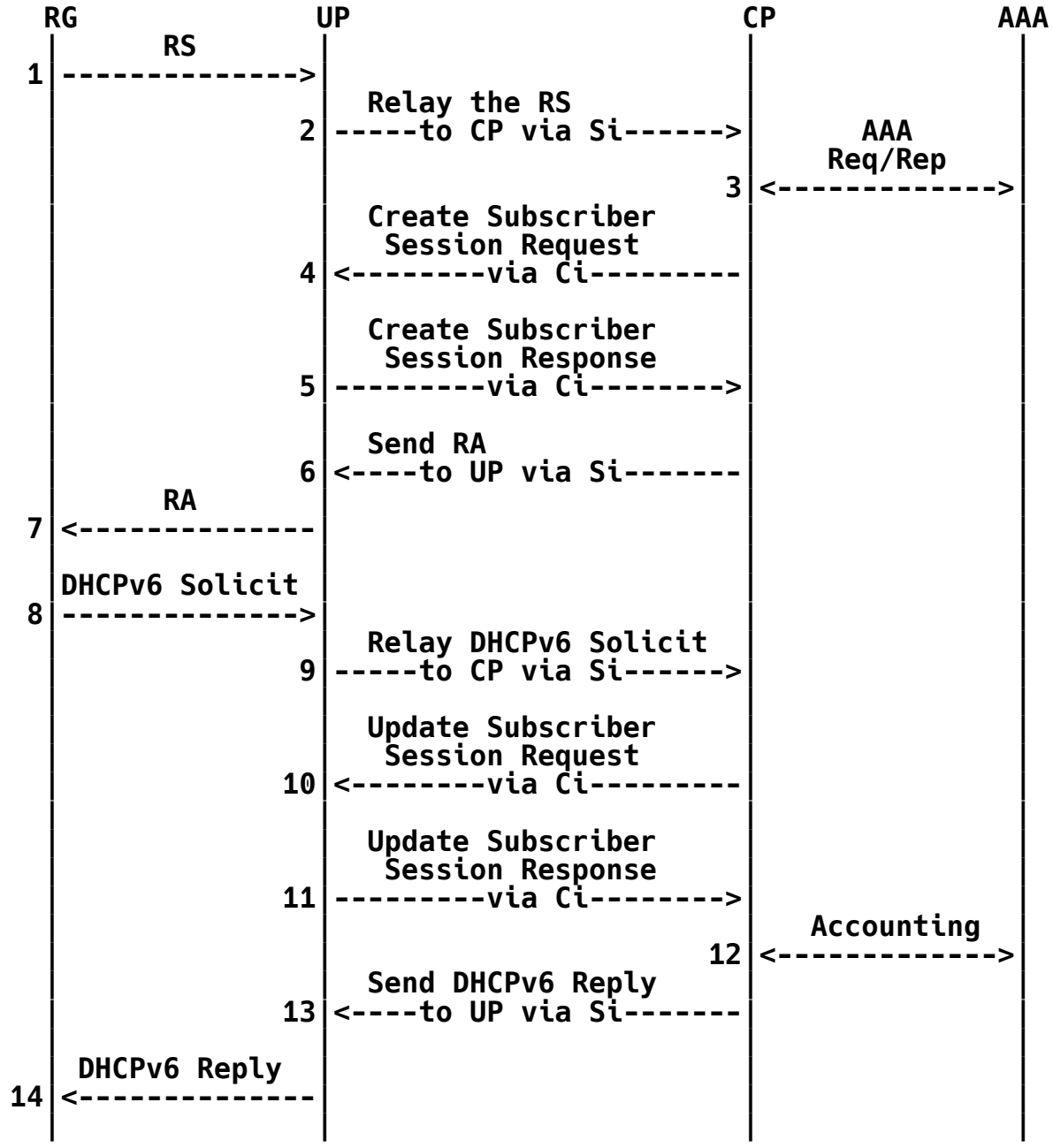


Figure 17: DHCPv6 and SLAAC Access

When a subscriber passes AAA authentication, the CP will create a subscriber session on the UP. This is achieved by sending an Update_Request message to the UP (step 4).

```
<Update_Request Message> ::= <Common Header>
                  <Basic Subscriber TLV>
                  <IPv6 Subscriber TLV>
                  <IPv6 Routing TLV>
                  [<Subscriber Policy TLV>]
```

The UP will reply with an Update_Response message (step 5). The format of the Update_Response is as follows:

<Update_Response Message> ::= <Common Header>
<Update Response TLV>

After receiving a DHCPv6 Solicit, the CP will update the subscriber session by sending an Update_Request message with new parameters to the UP (step 10).

The format of the Update_Request message is as follows:

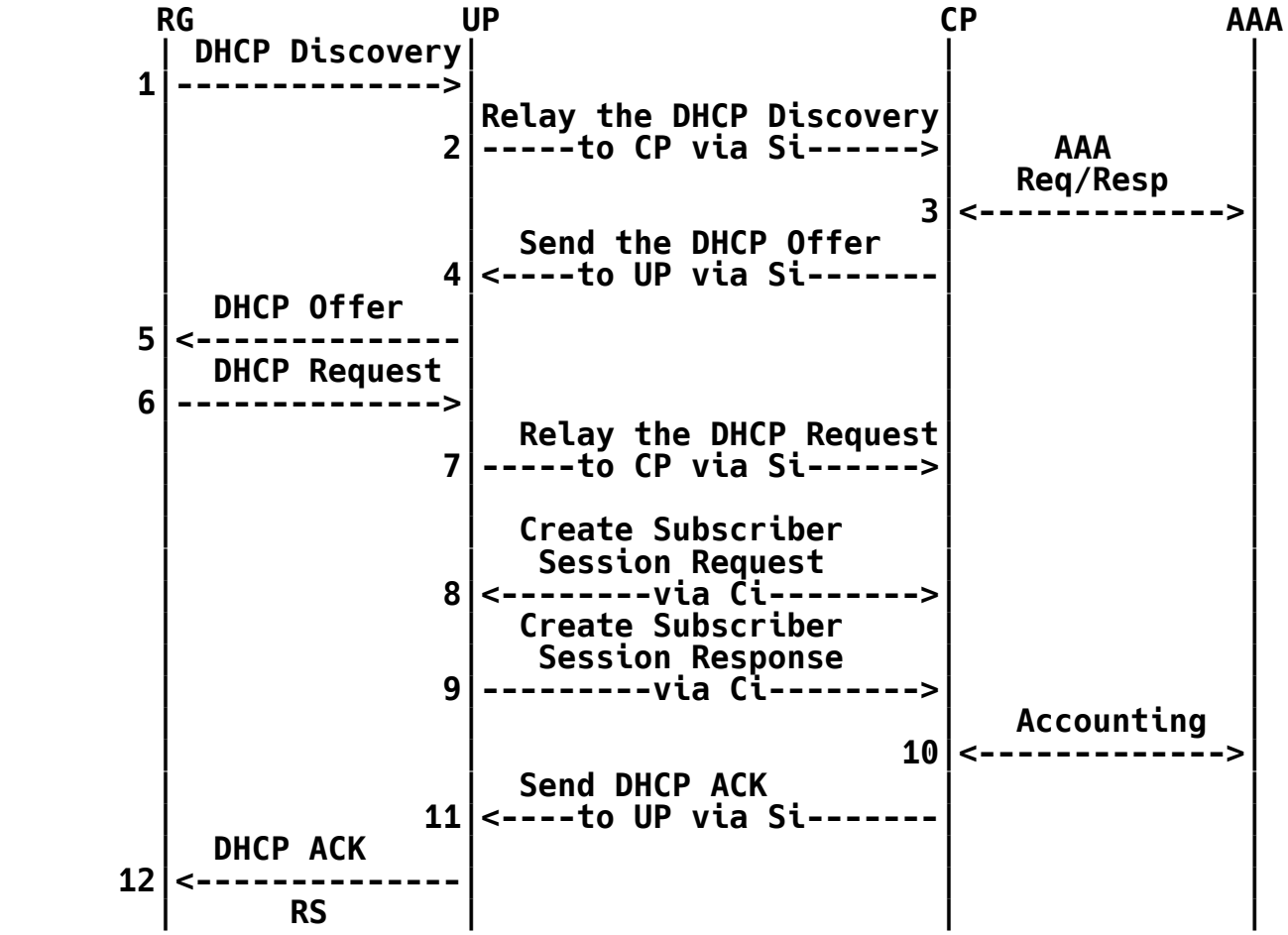
<Update_Request Message> ::= <Common Header>
<Basic Subscriber TLV>
<IPv6 Subscriber TLV>
<IPv6 Routing TLV>
[<Subscriber Policy TLV>]

The UP will reply with an Update_Response message (step 11). The format of the Update_Response is as follows:

<Update_Response Message> ::= <Common Header>
<Update Response TLV>

5.1.5. DHCP Dual-Stack Access

The following sequence (Figure 18) is a combination of DHCP IPv4 and DHCP IPv6 access processes.



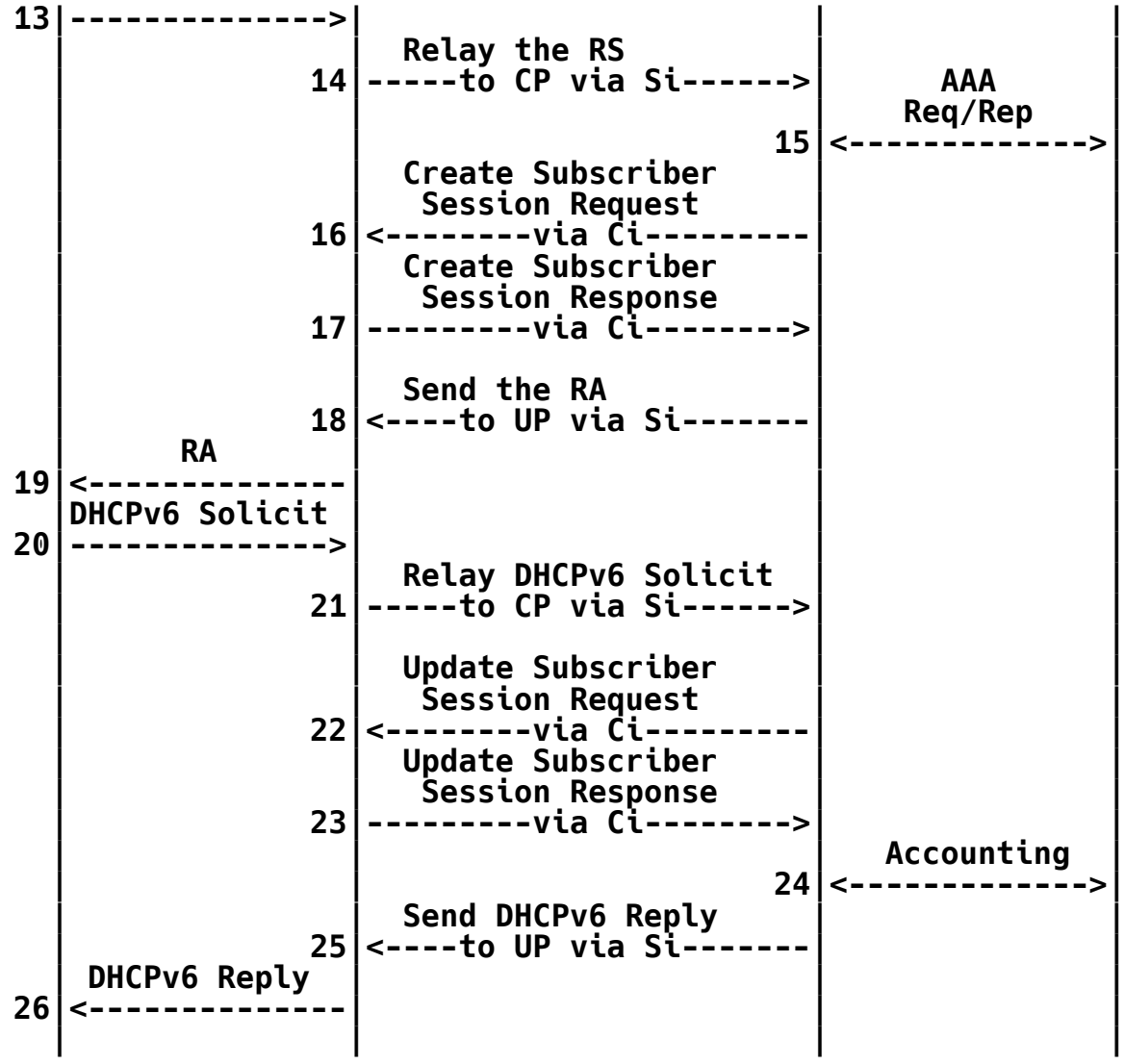


Figure 18: DHCP Dual-Stack Access

The DHCP dual-stack access includes three sets of Update_Request/ Update_Response exchanges to create/update a DHCPv4/v6 subscriber session.

(1) Create a DHCPv4 session (steps 8 and 9):

```
<Update_Request Message> ::= <Common Header>
                  <Basic Subscriber TLV>
                  <IPv4 Subscriber TLV>
                  <IPv4 Routing TLV>
                  [<Subscriber Policy TLV>]

<Update_Response Message> ::= <Common Header>
                 <Update Response TLV>
                 [<Subscriber CGN Port Range TLV>]
```

(2) Create a DHCPv6 session (steps 16 and 17):

```
<Update_Request Message> ::= <Common Header>
```

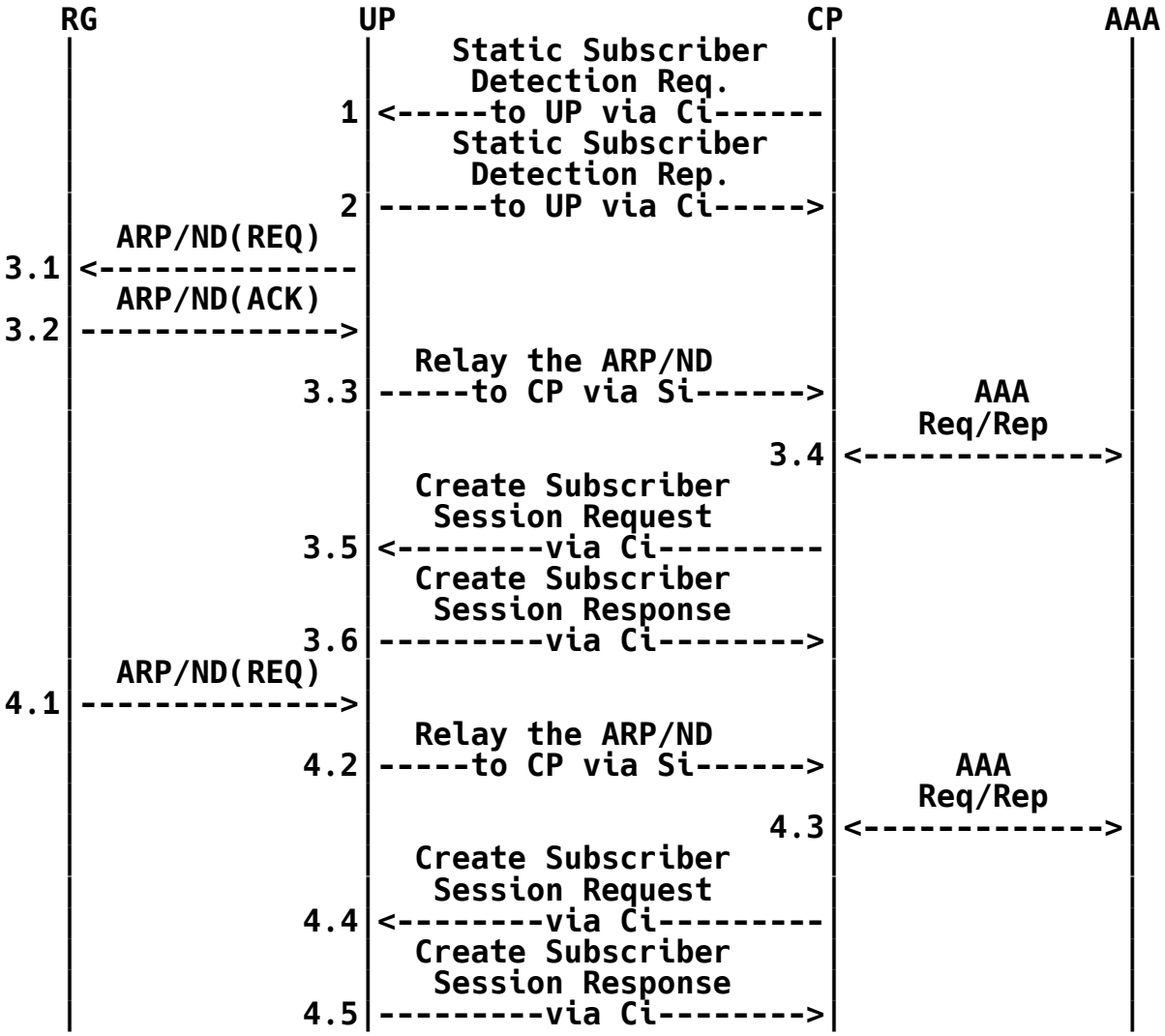
```
                        <Basic Subscriber TLV>
                        <IPv6 Subscriber TLV>
                        <IPv6 Routing TLV>
                        [<Subscriber Policy TLV>]

      <Update_Response Message> ::= <Common Header>
                       <Update Response TLV>
```

(3) Update DHCPv6 session (steps 22 and 23):

```
      <Update_Request Message> ::= <Common Header>
                        <Basic Subscriber TLV>
                        <IPv6 Subscriber TLV>
                        <IPv6 Routing TLV>
                        [<Subscriber Policy TLV>]

      <Update_Response Message> ::= <Common Header>
                       <Update Response TLV>
```

5.1.6. L2 Static Subscriber Access

L2 static subscriber access processes are as follows:



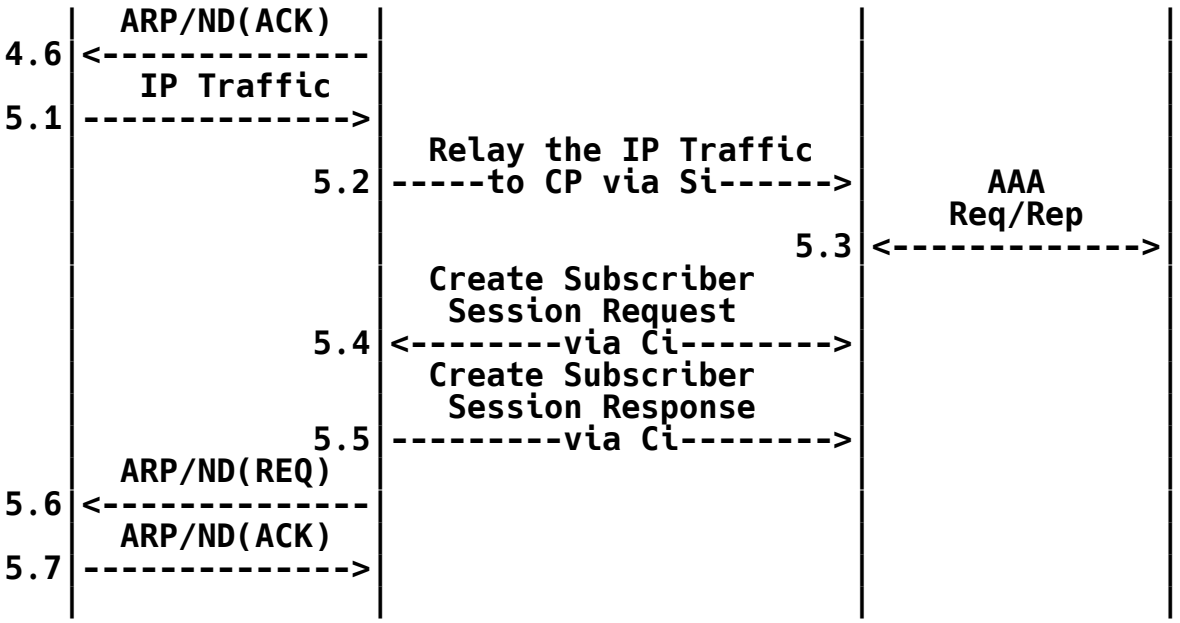


Figure 19: L2 Static Subscriber Access

For L2 static subscriber access, the process starts with a CP installing a static subscriber detection list on a UP. The list determines which subscribers will be detected. That is implemented by exchanging Update_Request and Update_Response messages between CP and UP. The formats of the messages are as follows:

```
<Update_Request Message> ::= <Common Header>
                  <IPv4 Static Subscriber Detect TLVs>
                  <IPv6 Static Subscriber Detect TLVs>

<Update_Response Message> ::= <Common Header>
                 <Update Response TLV>
```

For L2 static subscriber access, there are three ways to trigger the access process:

(1) Triggered by UP (steps 3.1-3.6): This assumes that the UP knows the IP address, the access interface, and the VLAN of the RG. The UP will actively trigger the access flow by sending an ARP/ND packet to the RG. If the RG is online, it will reply with an ARP/ND to the UP. The UP will tunnel the ARP/ND to the CP through the Si. The CP then triggers the authentication process. If the authentication result is positive, the CP will create a corresponding subscriber session on the UP.

(2) Triggered by RG ARP/ND (steps 4.1-4.6): Most of the process is the same as option 1 (triggered by UP). The difference is that the RG will actively send the ARP/ND to trigger the process.

(3) Triggered by RG IP traffic (steps 5.1-5.7): This is for the case where the RG has the ARP/ND information, but the subscriber session on the UP is lost (e.g., due to failure on the UP or the UP restarting). That means the RG may keep sending IP packets to the UP. The packets will trigger the UP to start a new

access process.

From a subscriber session point of view, the procedures and the message formats for the three cases above are the same, as follows.

IPv4 Case:

```
<Update_Request Message> ::= <Common Header>
                   <Basic Subscriber TLV>
                   <IPv4 Subscriber TLV>
                   <IPv4 Routing TLV>
                   [<Subscriber Policy TLV>]

<Update_Response Message> ::= <Common Header>
                  <Update Response TLV>
                  [<Subscriber CGN Port Range TLV>]
```

IPv6 Case:

```
<Update_Request Message> ::= <Common Header>
                   <Basic Subscriber TLV>
                   <IPv6 Subscriber TLV>
                   <IPv6 Routing TLV>
                   [<Subscriber Policy TLV>]

<Update_Response Message> ::= <Common Header>
                  <Update Response TLV>
```

5.2. PPPoE

5.2.1. IPv4 PPPoE Access

Figure 20 shows the IPv4 PPPoE access call flow.

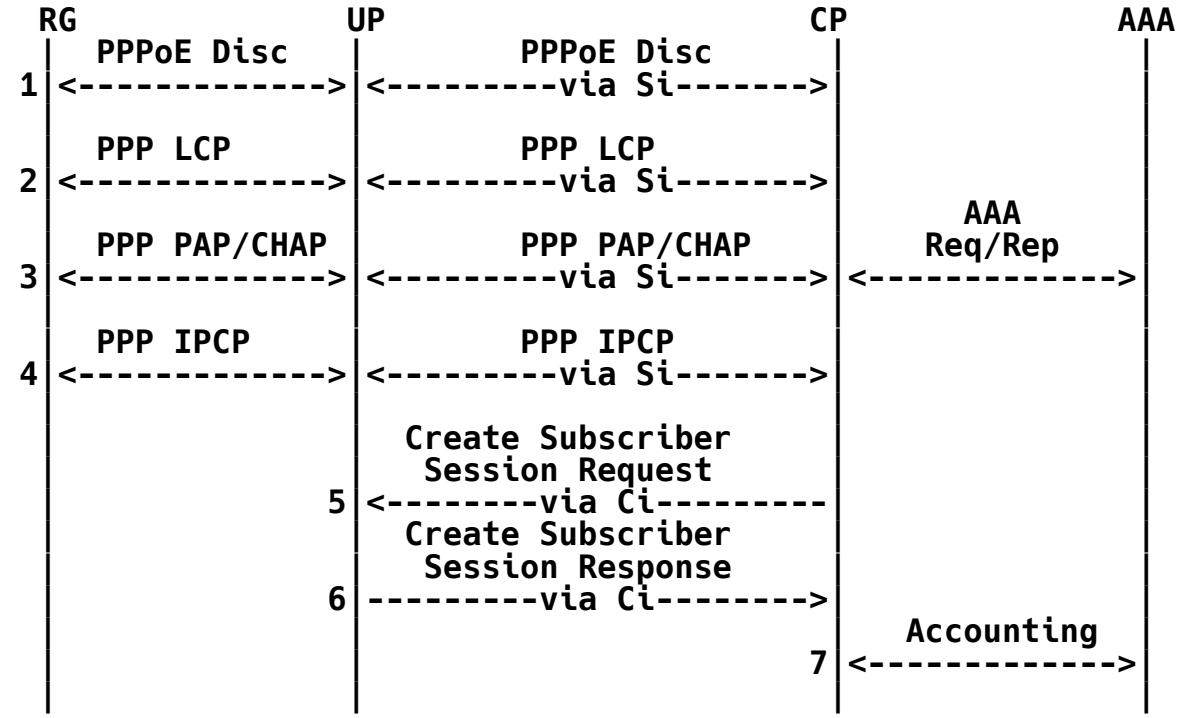


Figure 20: IPv4 PPPoE Access

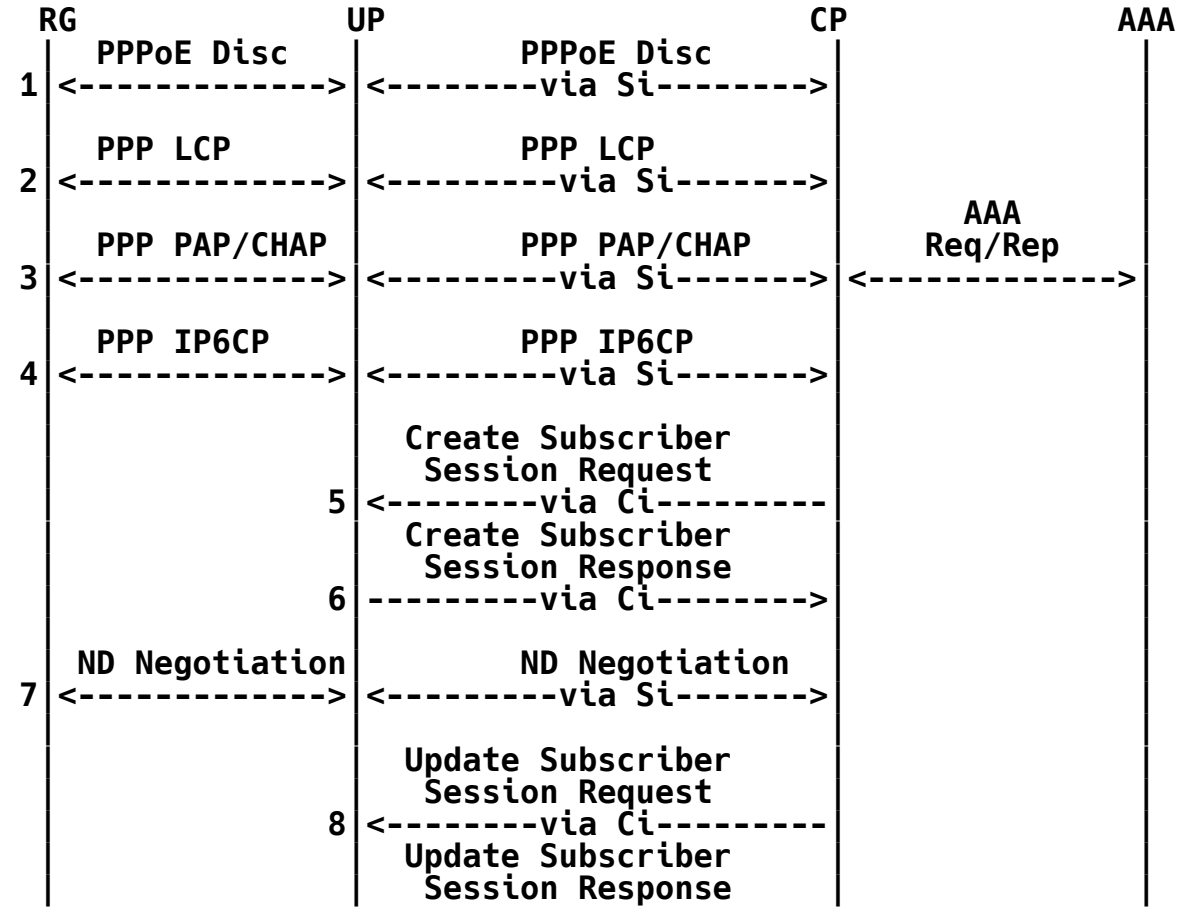
In the above sequence, steps 1-4 are the standard PPPoE call flow. The UP is responsible for redirecting the PPPoE control packets to the CP or RG. The PPPoE control packets are transmitted between the CP and UP through the Si.

After the PPPoE call flow, if the subscriber passed the AAA authentication and authorization, the CP will create a corresponding session on the UP through the Ci. The formats of the messages are as follows:

```
<Update_Request Message> ::= <Common Header>
                    <Basic Subscriber TLV>
                    <PPP Subscriber TLV>
                    <IPv4 Subscriber TLV>
                    <IPv4 Routing TLV>
                    [<Subscriber Policy TLV>]

<Update_Response Message> ::= <Common Header>
                   <Update Response TLV>
                   [<Subscriber CGN Port Range TLV>]
```

5.2.2. IPv6 PPPoE Access

Figure 21 describes the IPv6 PPPoE access call flow.



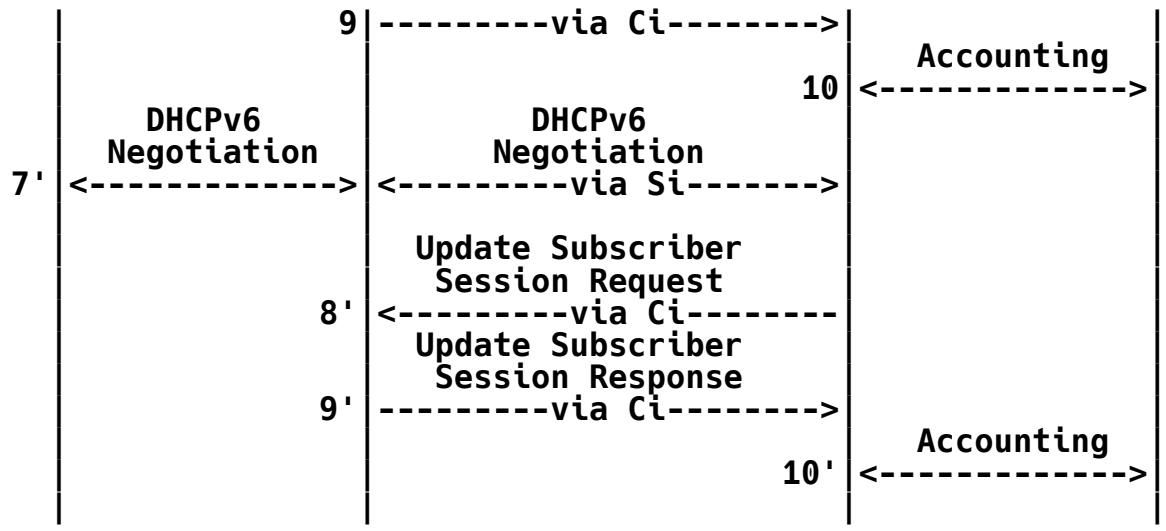


Figure 21: IPv6 PPPoE Access

From the above sequence, steps 1-4 are the standard PPPoE call flow. The UP is responsible for redirecting the PPPoE control packets to the CP or RG. The PPPoE control packets are transmitted between the CP and UP through the Si.

After the PPPoE call flow, if the subscriber passed the AAA authentication and authorization, the CP will create a corresponding session on the UP through the Ci. The formats of the messages are as follows:

```
<Update_Request Message> ::= <Common Header>
                  <Basic Subscriber TLV>
                  <PPP Subscriber TLV>
                  <IPv6 Subscriber TLV>
                  <IPv6 Routing TLV>
                  [<Subscriber Policy TLV>]

<Update_Response Message> ::= <Common Header>
                 <Update Response TLV>
```

Then, the RG will initialize an ND/DHCPv6 negotiation process with the CP (see steps 7 and 7'); after that, it will trigger an update (steps 8-9 and 8'-9') to the subscriber session. The formats of the update messages are as follows:

```
<Update_Request Message> ::= <Common Header>
                  <Basic Subscriber TLV>
                  <PPP Subscriber TLV>
                  <IPv6 Subscriber TLV>
                  <IPv6 Routing TLV>
                  [<Subscriber Policy TLV>]

<Update_Response Message> ::= <Common Header>
                 <Update Response TLV>
```

5.2.3. PPPoE Dual-Stack Access

Figure 22 shows a combination of IPv4 and IPv6 PPPoE access call flows.

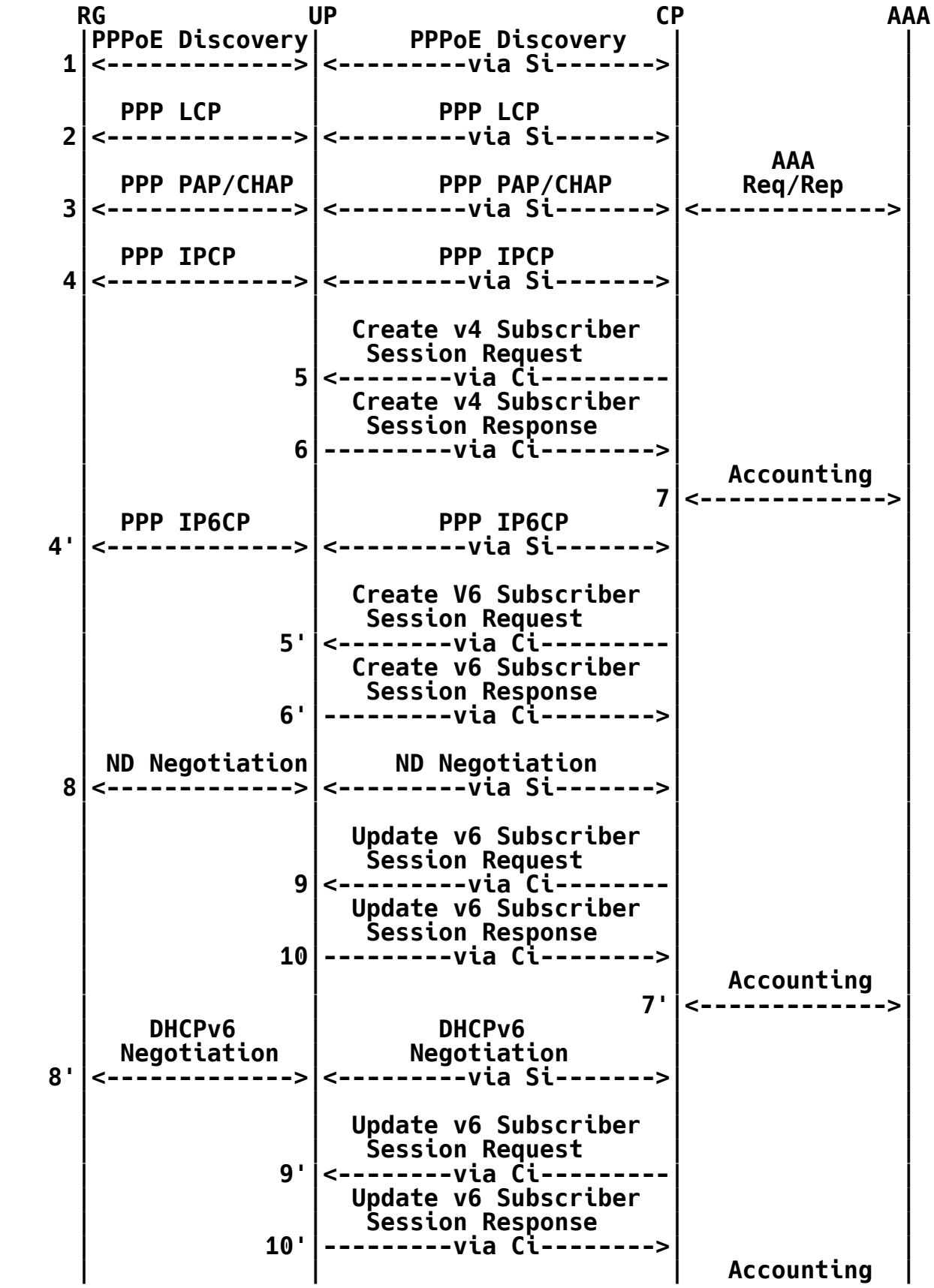


Figure 22: PPPoE Dual-Stack Access

PPPoE dual stack is a combination of IPv4 PPPoE and IPv6 PPPoE access. The process is as above. The formats of the messages are as follows:

(1) Create an IPv4 PPPoE subscriber session (steps 5-6):

```
<Update_Request Message> ::= <Common Header>
                  <Basic Subscriber TLV>
                  <PPP Subscriber TLV>
                  <IPv4 Subscriber TLV>
                  <IPv4 Routing TLV>
                  [<Subscriber Policy TLV>]

<Update_Response Message> ::= <Common Header>
                 <Update Response TLV>
                 [<Subscriber CGN Port Range TLV>]
```

(2) Create an IPv6 PPPoE subscriber session (steps 5'-6'):

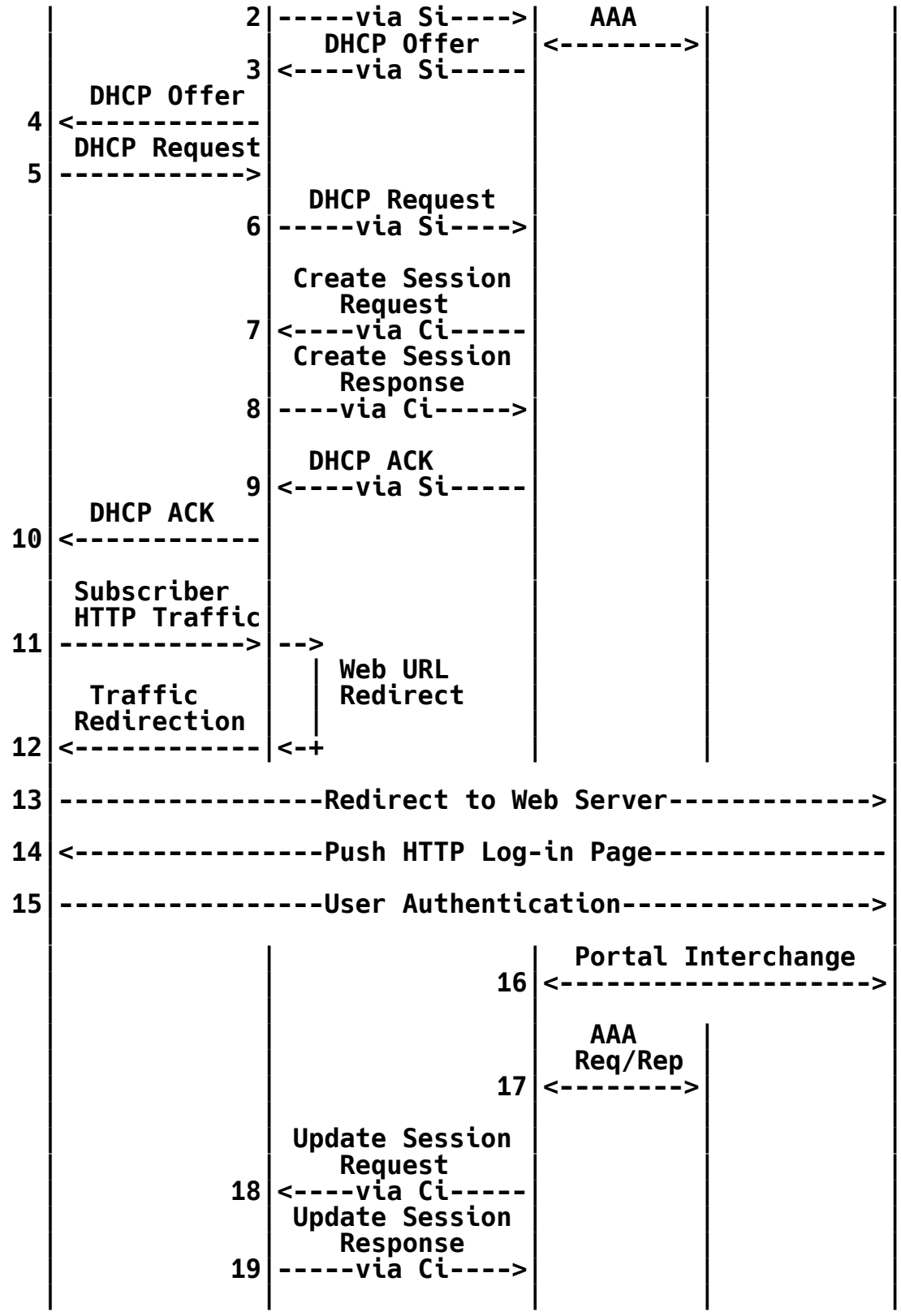
```
<Update_Request Message> ::= <Common Header>
                  <Basic Subscriber TLV>
                  <PPP Subscriber TLV>
                  <IPv6 Subscriber TLV>
                  <IPv6 Routing TLV>
                  [<Subscriber Policy TLV>]

<Update_Response Message> ::= <Common Header>
                 <Update Response TLV>
```

(3) Update the IPv6 PPPoE subscriber session (steps 9-10 and 9'-10'):

```
<Update_Request Message> ::= <Common Header>
                  <Basic Subscriber TLV>
                  <PPP Subscriber TLV>
                  <IPv6 Subscriber TLV>
                  <IPv6 Routing TLV>
                  [<Subscriber Policy TLV>]

<Update_Response Message> ::= <Common Header>
                 <Update Response TLV>
```

5.3. WLAN Access

Figure 23 shows the WLAN access call flow.



WLAN access starts with the DHCP dial-up process (steps 1-6). After that, the CP will create a subscriber session on the UP (steps 7-8). The formats of the session creation messages are as follows:

IPv4 Case:

```
<Update_Request Message> ::= <Common Header>
                  <Basic Subscriber TLV>
                  <IPv4 Subscriber TLV>
                  <IPv4 Routing TLV>
                  [<Subscriber Policy TLV>]

<Update_Response Message> ::= <Common Header>
                 <Update Response TLV>
                 [<Subscriber CGN Port Range TLV>]
```

IPv6 Case:

```
<Update_Request Message> ::= <Common Header>
                  <Basic Subscriber TLV>
                  <IPv6 Subscriber TLV>
                  <IPv6 Routing TLV>
                  [<Subscriber Policy TLV>]

<Update_Response Message> ::= <Common Header>
                 <Update Response TLV>
```

After step 10, the RG will be allocated an IP address, and its first HTTP packet will be redirected to a web server for subscriber authentication (steps 11-17). After the web authentication, if the result is positive, the CP will update the subscriber session by using the following message exchanges:

IPv4 Case:

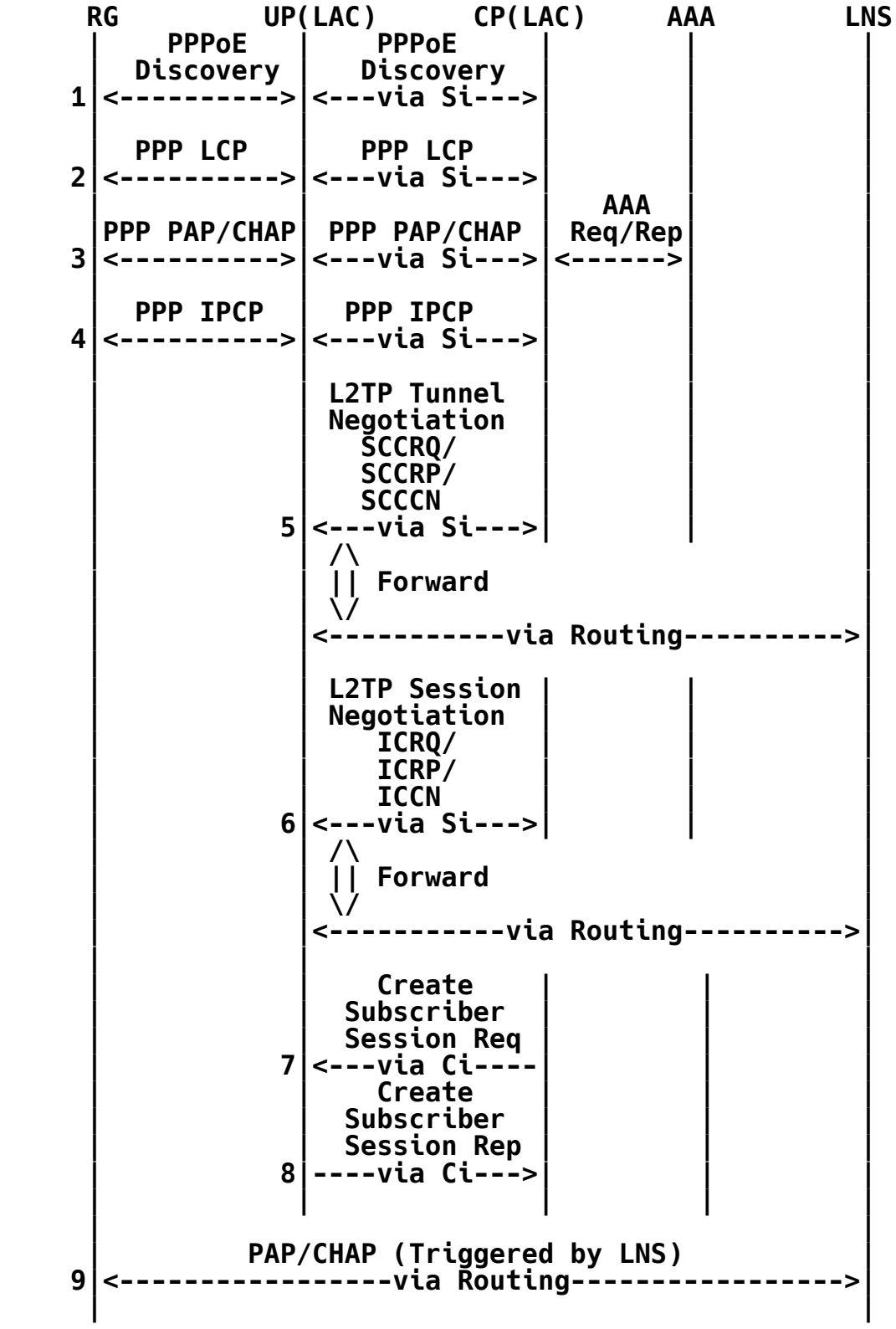
```
<Update_Request Message> ::= <Common Header>
                  <Basic Subscriber TLV>
                  <IPv4 Subscriber TLV>
                  <IPv4 Routing TLV>
                  [<Subscriber Policy TLV>]

<Update_Response Message> ::= <Common Header>
                 <Update Response TLV>
                 [<Subscriber CGN Port Range TLV>]
```

IPv6 Case:

```
<Update_Request Message> ::= <Common Header>
                  <Basic Subscriber TLV>
                  <IPv6 Subscriber TLV>
                  <IPv6 Routing TLV>
                  [<Subscriber Policy TLV>]

<Update_Response Message> ::= <Common Header>
                 <Update Response TLV>
```

5.4. L2TP

5.4.1. L2TP LAC Access



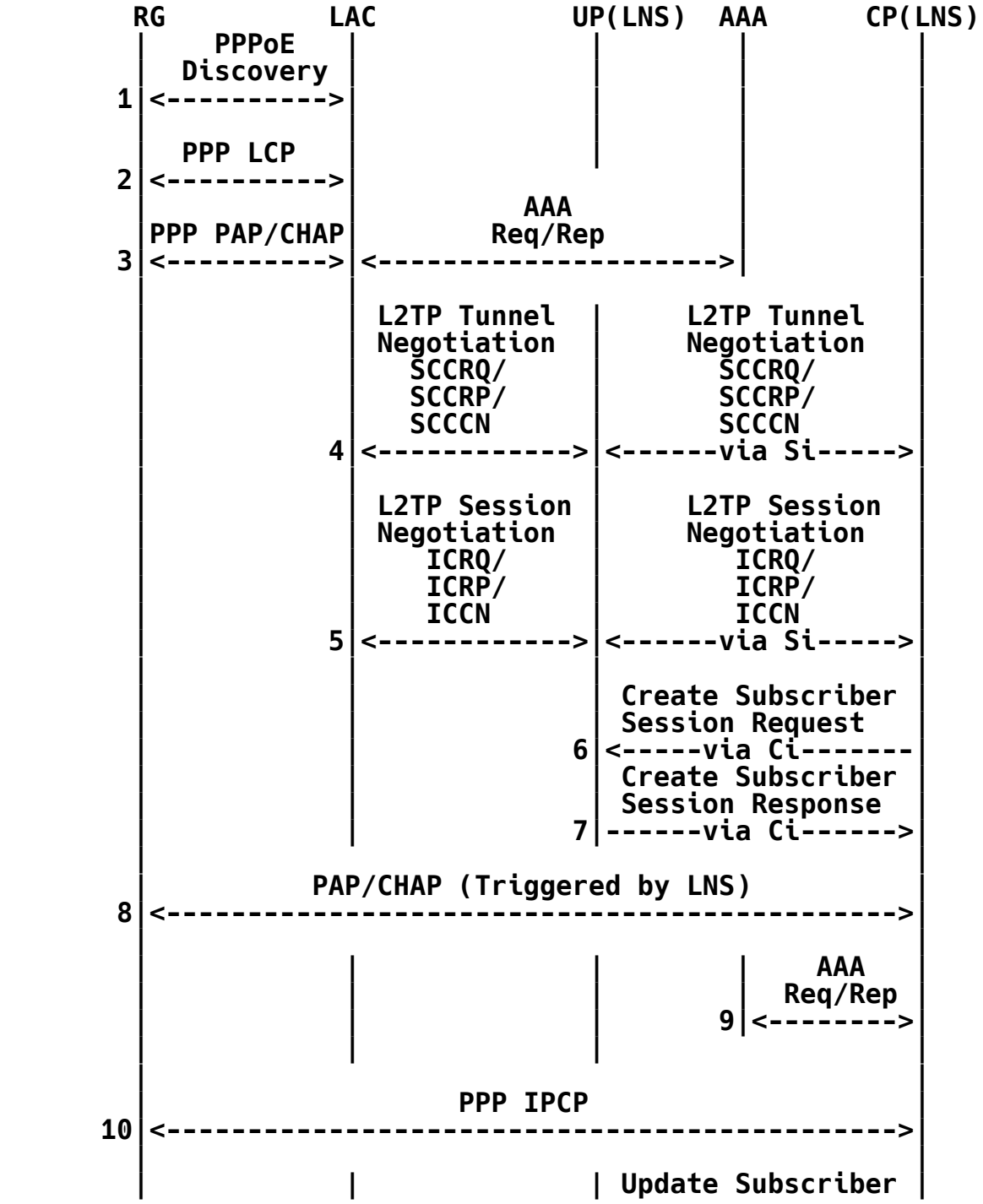
Steps 1-4 are a standard PPPoE access process. After that, the LAC-CP starts to negotiate an L2TP session and tunnel with the LNS. After the negotiation, the CP will create an L2TP LAC subscriber

session on the UP through the following messages:

<Update_Request Message> ::= <Common Header>
<Basic Subscriber TLV>
<L2TP-LAC Subscriber TLV>
<L2TP-LAC Tunnel TLV>

<Update_Response Message> ::= <Common Header>
<Update Response TLV>

5.4.2. L2TP LNS IPv4 Access



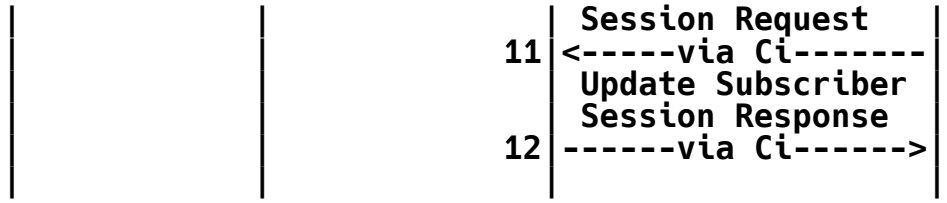


Figure 25: L2TP LNS IPv4 Access

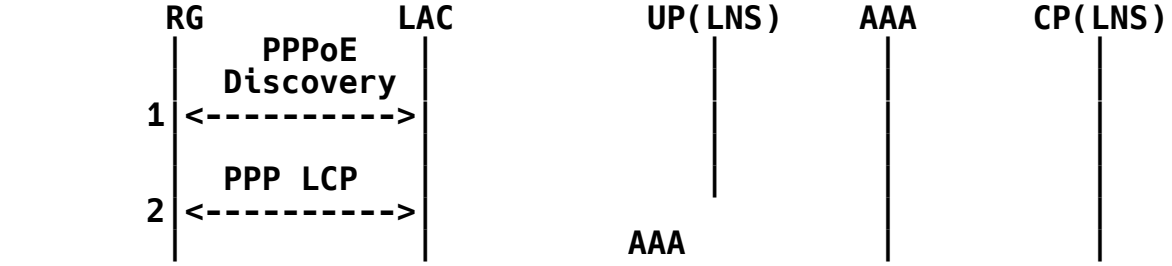
In this case, the BNG is running as an LNS and separated into LNS-CP and LNS-UP. Steps 1-5 finish the normal L2TP dial-up process. When the L2TP session and tunnel negotiations are finished, the LNS-CP will create an L2TP LNS subscriber session on the LNS-UP. The format of the messages is as follows:

```
<Update_Request Message> ::= <Common Header>
                  <L2TP-LNS Subscriber TLV>
                  <Basic Subscriber TLV>
                  <PPP Subscriber TLV>
                  <IPv4 Subscriber TLV>
                  <IPv4 Routing TLV>
                  <L2TP-LNS Tunnel TLV>
                  [<Subscriber Policy TLV>]

<Update_Response Message> ::= <Common Header>
                 <Update Response TLV>
                 [<Subscriber CGN Port Range TLV>]
```

After that, the LNS-CP will trigger a AAA authentication. If the authentication result is positive, a PPP IP Control Protocol (IPCP) process will follow, and then the CP will update the session with the following message exchanges:

```
<Update_Request Message> ::= <Common Header>
                  <L2TP-LNS Subscriber TLV>
                  <Basic Subscriber TLV>
                  <PPP Subscriber TLV>
                  <IPv4 Subscriber TLV>
                  <IPv4 Routing TLV>
                  <L2TP-LNS Tunnel TLV>
                  [<Subscriber Policy TLV>]

<Update_Response Message> ::= <Common Header>
                 <Update Response TLV>
                 [<Subscriber CGN Port Range TLV>]
```

5.4.3. L2TP LNS IPv6 Access



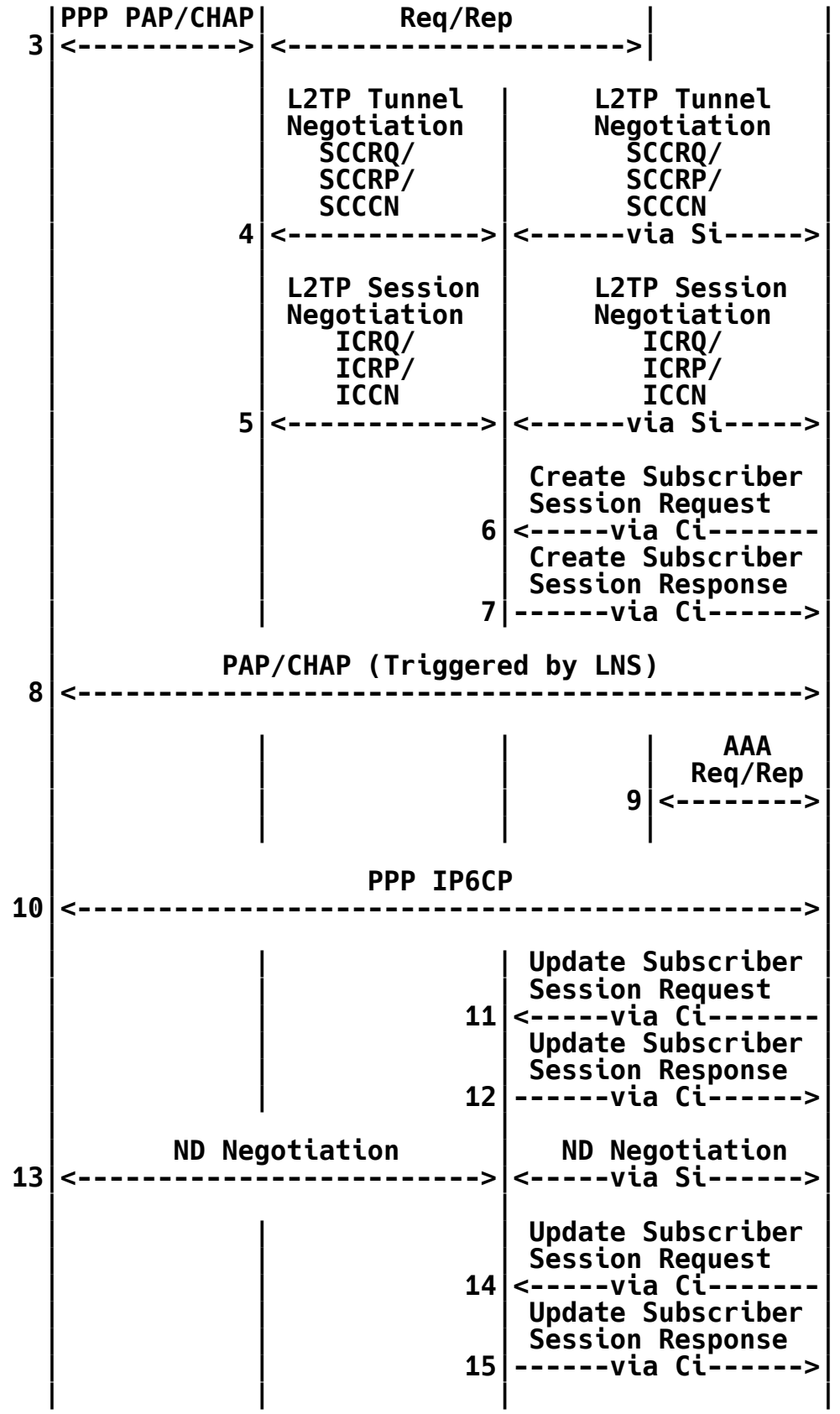


Figure 26: L2TP LNS IPv6 Access

Steps 1-12 are the same as L2TP LNS IPv4 access. Steps 1-5 finish

the normal L2TP dial-up process. When the L2TP session and tunnel negotiations are finished, the LNS-CP will create an L2TP LNS subscriber session on the LNS-UP. The format of the messages is as follows:

```
<Update_Request Message> ::= <Common Header>
                  <L2TP-LNS Subscriber TLV>
                  <Basic Subscriber TLV>
                  <PPP Subscriber TLV>
                  <IPv6 Subscriber TLV>
                  <IPv6 Routing TLV>
                  <L2TP-LNS Tunnel TLV>
                  [<Subscriber Policy TLV>]

<Update_Response Message> ::= <Common Header>
                 <Update Response TLV>
```

After that, the LNS-CP will trigger a AAA authentication. If the authentication result is positive, a PPP IP6CP process will follow, and then the CP will update the session with the following message exchanges:

```
<Update_Request Message> ::= <Common Header>
                  <L2TP-LNS Subscriber TLV>
                  <Basic Subscriber TLV>
                  <PPP Subscriber TLV>
                  <IPv6 Subscriber TLV>
                  <IPv6 Routing TLV>
                  <L2TP-LNS Tunnel TLV>
                  [<Subscriber Policy TLV>]

<Update_Response Message> ::= <Common Header>
                 <Update Response TLV>
```

Then, an ND negotiation will be triggered by the RG. After the ND negotiation, the CP will update the session with the following message exchanges:

```
<Update_Request Message> ::= <Common Header>
                  <L2TP-LAC Subscriber TLV>
                  <Basic Subscriber TLV>
                  <PPP Subscriber TLV>
                  <IPv6 Subscriber TLV>
                  <IPv6 Routing TLV>
                  <L2TP-LNS Tunnel TLV>
                  [<Subscriber Policy TLV>]

<Update_Response Message> ::= <Common Header>
                 <Update Response TLV>
```

5.5. CGN (Carrier Grade NAT)

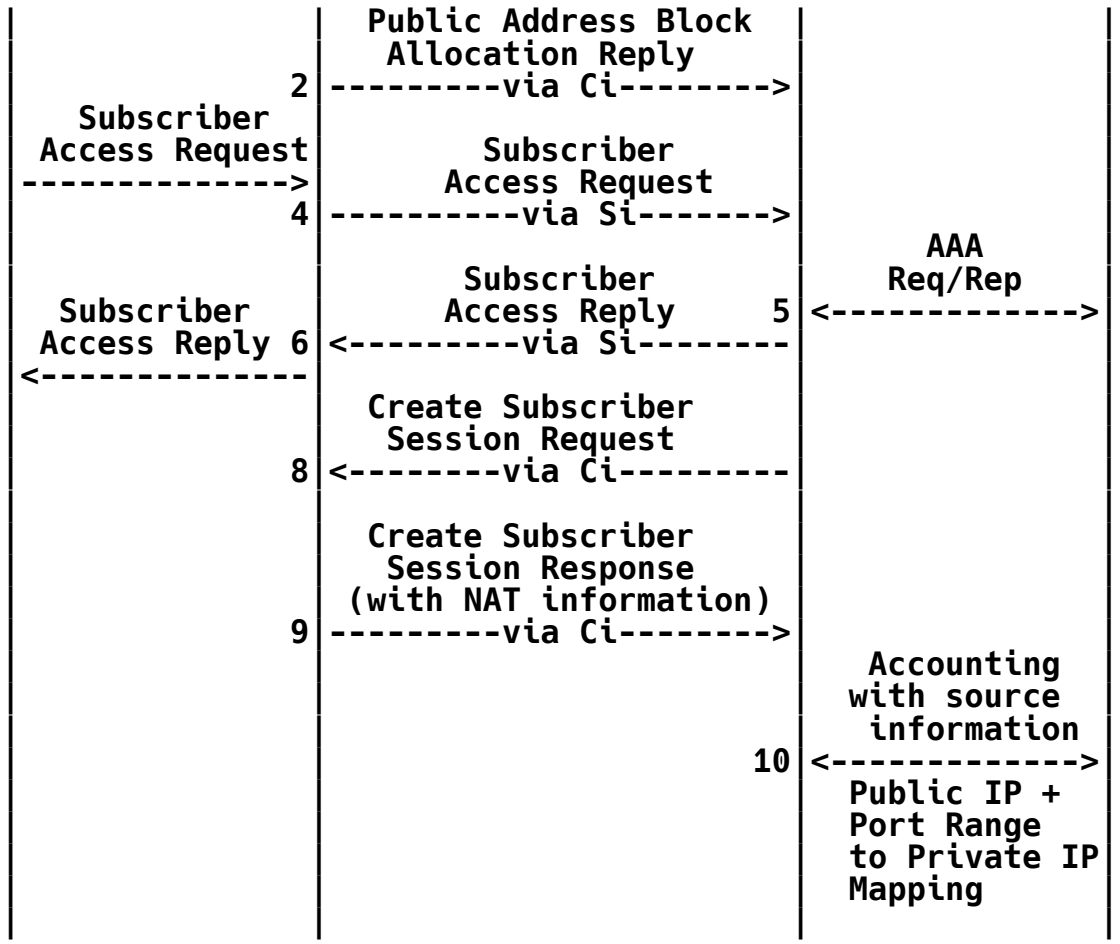


Figure 27: CGN Access

The first steps allocate one or more CGN address blocks to the UP (steps 1-2). This is achieved by the following message exchanges between CP and UP:

<Addr_Allocation_Req Message> ::= <Common Header>
<Address Allocation Request TLV>

<Addr_Allocation_Ack Message> ::= <Common Header>
<Address Allocation Response TLV>

Steps 3-9 show the general dial-up process in the case of CGN mode. The specific processes (e.g., IPoE, PPPoE, L2TP, etc.) are defined in above sections.

If a subscriber is a CGN subscriber, once the subscriber session is created/updated, the UP will report the NAT information to the CP. This is achieved by carrying the Subscriber CGN Port Range TLV in the Update_Response message.

5.6. L3 Leased Line Access

5.6.1. Web Authentication

RG UP CP AAA Web Server

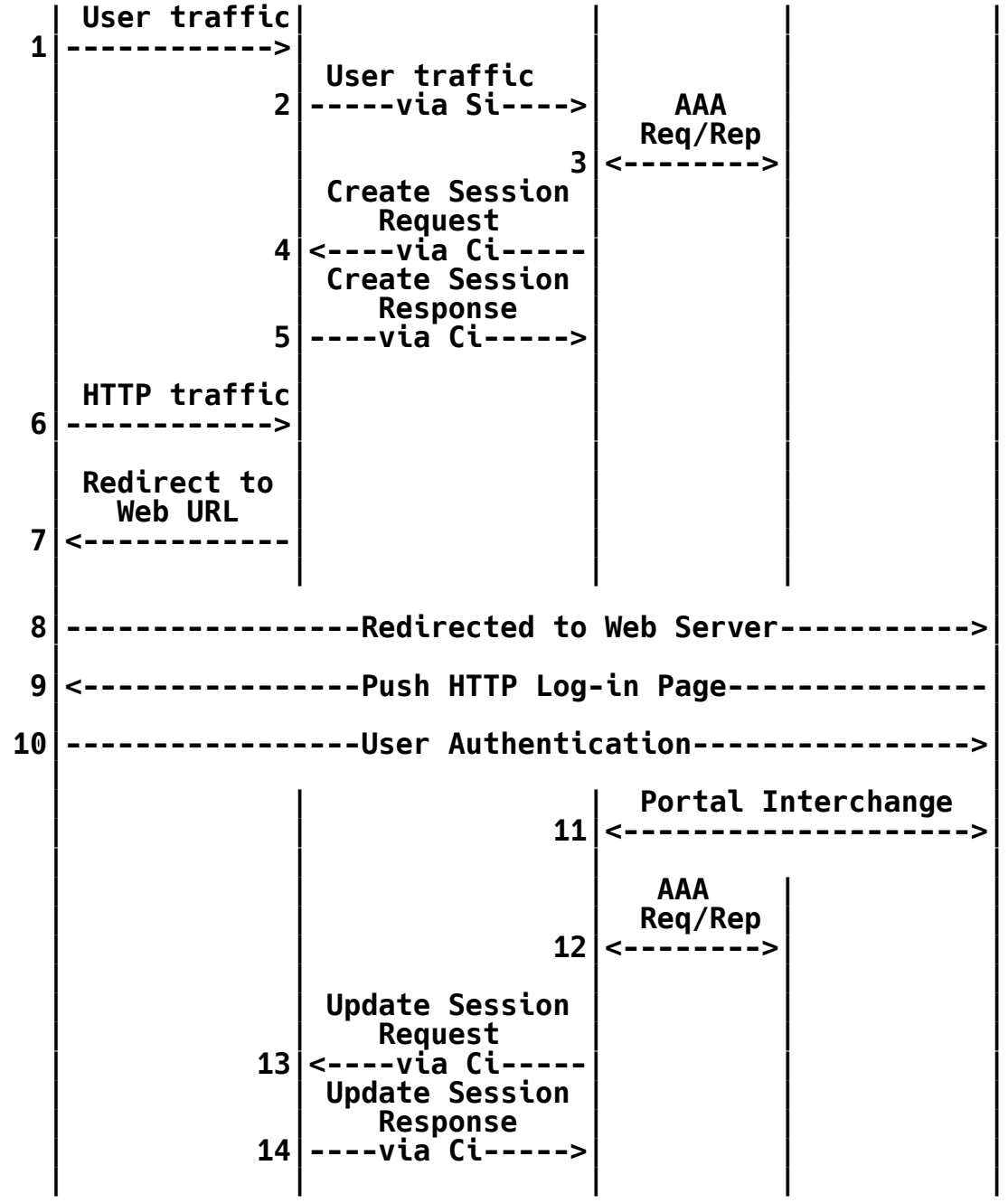


Figure 28: Web Authentication-Based L3 Leased Line Access

In this case, IP traffic from the RG will trigger the CP to authenticate the RG by checking the source IP and the exchanges with the AAA server. Once the RG has passed the authentication, the CP will create a corresponding subscriber session on the UP through the following message exchanges:

IPv4 Case:

```
<Update_Request Message> ::= <Common Header>
                  <Basic Subscriber TLV>
                  <IPv4 Subscriber TLV>
                  <IPv4 Routing TLV>
```

[<Subscriber Policy TLV>]

<Update_Response Message> ::= <Common Header>

<Update Response TLV>
[<Subscriber CGN Port Range TLV>]

IPv6 Case:

<Update_Request Message> ::= <Common Header>
<Basic Subscriber TLV>
<IPv6 Subscriber TLV>
<IPv6 Routing TLV>
[<Subscriber Policy TLV>]

<Update_Response Message> ::= <Common Header>
<Update Response TLV>

Then, the HTTP traffic from the RG will be redirected to a web server to finish the web authentication. Once the web authentication is passed, the CP will trigger another AAA authentication. After the AAA authentication, the CP will update the session with the following message exchanges:

IPv4 Case:

<Update_Request Message> ::= <Common Header>
<Basic Subscriber TLV>
<IPv4 Subscriber TLV>
<IPv4 Routing TLV>
[<Subscriber Policy TLV>]

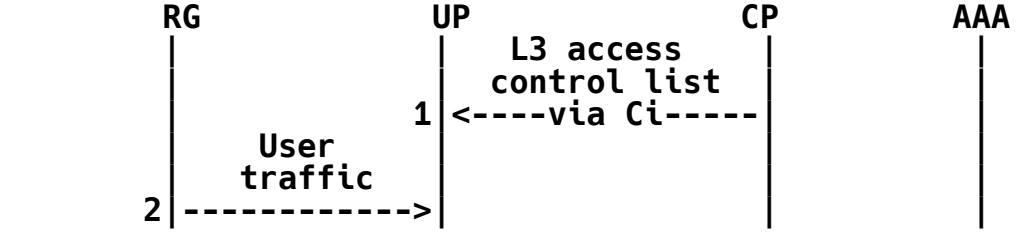
<Update_Response Message> ::= <Common Header>
<Update Response TLV>
[<Subscriber CGN Port Range TLV>]

IPv6 Case:

<Update_Request Message> ::= <Common Header>
<Basic Subscriber TLV>
<IPv6 Subscriber TLV>
<IPv6 Routing TLV>
[<Subscriber Policy TLV>]

<Update_Response Message> ::= <Common Header>
<Update Response TLV>

5.6.2. User Traffic Trigger



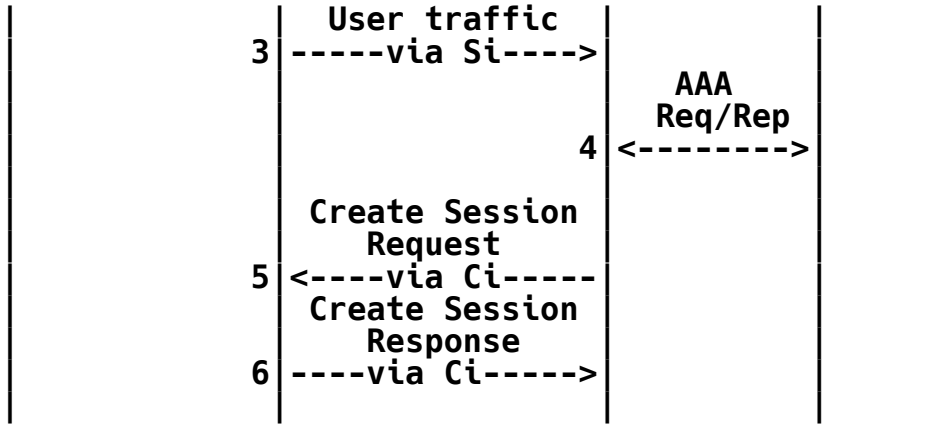


Figure 29: User Traffic Triggered L3 Leased Line Access

In this case, the CP must install on the UP an access control list, which is used by the UP to determine whether or not an RG is legal. If the traffic is from a legal RG, it will be redirected to the CP though the Si. The CP will trigger a AAA interchange with the AAA server. After that, the CP will create a corresponding subscriber session on the UP with the following message exchanges:

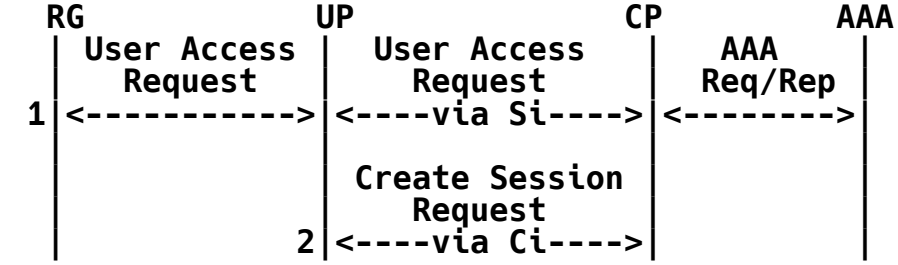
IPv4 Case:

```
<Update_Request Message> ::= <Common Header>
                  <Basic Subscriber TLV>
                  <IPv4 Subscriber TLV>
                  <IPv4 Routing TLV>
                  [<Subscriber Policy TLV>]

<Update_Response Message> ::= <Common Header>
                 <Update Response TLV>
                 [<Subscriber CGN Port Range TLV>]
```

IPv6 Case:

```
<Update_Request Message> ::= <Common Header>
                  <Basic Subscriber TLV>
                  <IPv6 Subscriber TLV>
                  <IPv6 Routing TLV>
                  [<Subscriber Policy TLV>]

<Update_Response Message> ::= <Common Header>
                 <Update Response TLV>
```

5.7. Multicast Service Access



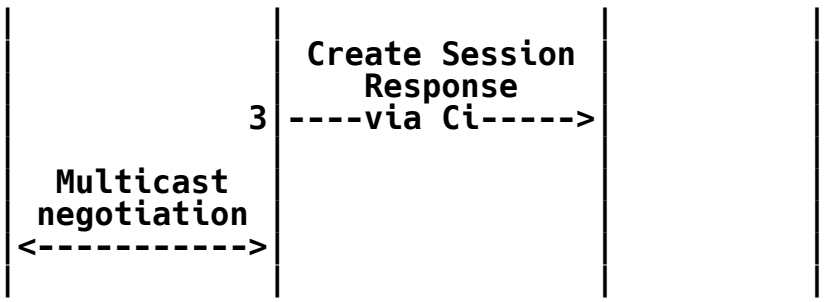


Figure 30: Multicast Access

Multicast access starts with a user access request from the RG. The request will be redirected to the CP by the Si. A follow-up AAA interchange between the CP and the AAA server will be triggered. After the authentication, the CP will create a multicast subscriber session on the UP through the following messages:

IPv4 Case, there will be a Multicast-ProfileV4 sub-TLV present in the Subscriber Policy TLV:

```
<Update_Request Message> ::= <Common Header>
                <Basic Subscriber TLV>
                <IPv4 Subscriber TLV>
                <IPv4 Routing TLV>
                <Subscriber Policy TLV>

<Update_Response Message> ::= <Common Header>
               <Update Response TLV>
               [<Subscriber CGN Port Range TLV>]
```

IPv6 Case, there will be a Multicast-ProfileV6 sub-TLV present in the Subscriber Policy TLV:

```
<Update_Request Message> ::= <Common Header>
                <Basic Subscriber TLV>
                <IPv6 Subscriber TLV>
                <IPv6 Routing TLV>
                <Subscriber Policy TLV>

<Update_Response Message> ::= <Common Header>
               <Update Response TLV>
```

6. S-CUSP Message Formats

An S-CUSP message consists of a common header followed by a variable-length body consisting entirely of TLVs. Receiving an S-CUSP message with an unknown message type or missing mandatory TLV MUST trigger an Error message (see Section 6.7) or a Response message with an Error Information TLV (see Section 7.6).

Conversely, if a TLV is optional, the TLV may or may not be present. Optional TLVs are indicated in the message formats shown in this document by being enclosed in square brackets.

This section specifies the format of the common S-CUSP message header

and lists the defined messages.

Network byte order is used for all multi-byte fields.

6.1. Common Message Header

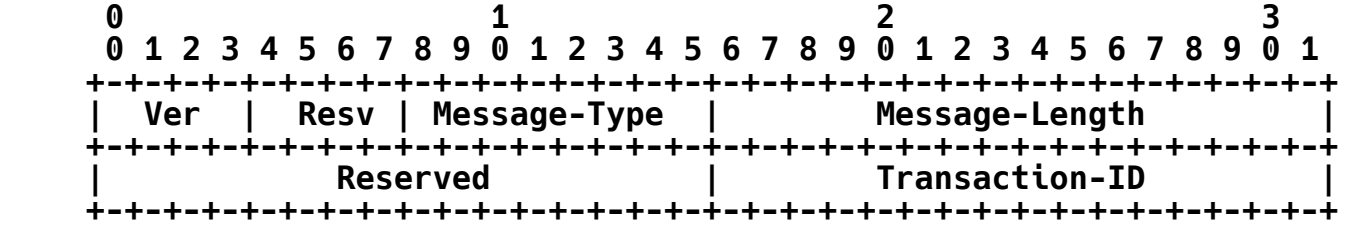


Figure 31: S-CUSP Message Common Header

Ver (4 bits): The major version of the protocol. This document specifies version 1. Different major versions of the protocol may have significantly different message structures and formats except that the Ver field will always be in the same place at the beginning of each message. A successful S-CUSP session depends on the CP and the UP both using the same major version of the protocol.

Resv (4 bits): Reserved. MUST be sent as zero and ignored on receipt.

Message-Type (8 bits): The set of message types specified in this document is listed in Section 8.1.

Message-Length (16 bits): Total length of the S-CUSP message including the common header, expressed in number of bytes as an unsigned integer.

Transaction-ID (16 bits): This field is used to identify requests. It is echoed back in any corresponding ACK/Response/Error message. It is RECOMMENDED that a monotonically increasing value be used in successive messages and that the value wraps back to zero after 0xFFFF. The content of this field is an opaque value that the receiver MUST NOT use for any purpose except to echo back in a corresponding response and, optionally, for logging.

6.2. Control Messages

This document defines the following control messages:

Type	Name	Notes and TLVs that can be carried
1	Hello	Hello TLV, Keepalive TLV
2	Keepalive	A common header with the Keepalive message type
3	Sync_Request	Synchronization request

4	Sync_Begin	Synchronization starts
5	Sync_Data	Synchronization data: TLVs specified in Section 7
6	Sync_End	End synchronization
7	Update_Request	TLVs specified in Sections 7.6-7.9
8	Update_Response	TLVs specified in Sections 7.6-7.9

Table 2: Control Messages

6.2.1. Hello Message

The Hello message is used for S-CUSP session establishment and version negotiation. The details of S-CUSP session establishment and version negotiation can be found in Section 4.1.1.

The format of the Hello message is as follows:

```
<Hello Message> ::= <Common Header>
                    <Hello TLV>
                    <Keepalive TLV>
                    [<Error Information TLV>]
```

The return code and negotiation result will be carried in the Error Information TLV. They are listed as follows:

0: Success. Version negotiation success.

1: Failure. Malformed message received.

2: TLV-Unknown. One or more of the TLVs was not understood.

1001: Version-Mismatch. The version negotiation fails. The S-CUSP session establishment phase fails.

1002: Keepalive Error. The keepalive negotiation fails. The S-CUSP session establishment phase fails.

1003: Timer Expires. The establishment timer expired. Session establishment phase fails.

6.2.2. Keepalive Message

Each end of an S-CUSP session periodically sends a Keepalive message. It is used to detect whether the peer end is still alive. The Keepalive procedures are defined in Section 4.1.2.

The format of the Keepalive message is as follows:

```
<Keepalive Message> ::= <Common Header>
```

6.2.3. Sync_Request Message

The Sync_Request message is used to request synchronization from an S-CUSP peer. Both CP and UP can request their peer to synchronize data.

The format of the Sync_Request message is as follows:

```
<Sync_Request Message> ::= <Common Header>
```

A Sync_Request message may result in a Sync_Begin message from its peer. The Sync_Begin message is defined in Section 6.2.4.

6.2.4. Sync_Begin Message

The Sync_Begin message is a reply to a Sync_Request message. It is used to notify the synchronization requester whether the synchronization can be started.

The format of the Sync_Begin message is as follows:

```
<Sync_Begin Message> ::= <Common Header>
                         <Error Information TLV>
```

The return codes are carried in the Error Information TLV. The codes are listed below:

0: Success. Be ready to synchronize.

1: Failure. Malformed message received.

2: TLV-Unknown. One or more of the TLVs was not understood.

2001: Synch-NoReady. The data to be synchronized is not ready.

2002: Synch-Unsupport. The data synchronization is not supported.

6.2.5. Sync_Data Message

The Sync_Data message is used to send data being synchronized between the CP and UP. The Sync_Data message has the same function and format as the Update_Request message. The difference is that there is no ACK for a Sync_Data message. An error caused by the Sync_Data message will result in a Sync_End message.

There are two scenarios:

* Synchronization from UP to CP: Synchronize the resource data to CP.

```
      <Sync_Data Message> ::= <Common Header>
                              [<Interface Status TLV>]
                              [<Board Status TLV>]
```

* Synchronization from CP to UP: Synchronize all subscriber sessions to the UP. The Subscriber TLVs carried are those appearing in Section 7.9. As for which TLVs should be carried, it depends on

the specific session data to be synchronized. The process is equivalent to the creation of a particular session. Refer to Section 5 to see more details.

```
<Sync_Data Message> ::= <Common Header>
                        [<IPv4 Routing TLV>]
                        [<IPv6 Routing TLV>]
                        [<Subscriber TLVs>]
```

6.2.6. Sync_End Message

The Sync_End message is used to indicate the end of a synchronization process. The format of a Sync_End message is as follows:

```
<Sync_End Message> ::= <Common Header>
                       <Error Information TLV>
```

The return/error codes are listed as follows:

0: Success. Synchronization finished.

1: Failure. Malformed message received.

2: TLV-Unknown. One or more of the TLVs was not understood.

6.2.7. Update_Request Message

The Update_Request message is a multipurpose message; it can be used to create, update, and delete subscriber sessions on a UP.

For session operations, the specific operation is controlled by the Oper field of the carried TLVs. As defined in Section 7.1, the Oper field can be set to either Update or Delete when a TLV is carried in an Update_Request message.

When the Oper field is set to Update, it means to create or update a subscriber session. If the Oper field is set to Delete, it is a request to delete a corresponding session.

The format of the Update_Request message is as follows:

```
<Update_Request Message> ::= <Common Header>
                         [<IPv4 Routing TLV>]
                         [<IPv6 Routing TLV>]
                         [<Subscriber TLVs>]
```

Where the Subscriber TLVs are those appearing in Section 7.9. Each Update_Request message will result in an Update_Response message, which is defined in Section 6.2.8.

6.2.8. Update_Response Message

The Update_Response message is a response to an Update_Request message. It is used to confirm the update request (or reject it in the case of an error). The format of an Update_Response message is as follows:

<Update_Response Message> ::= <Common Header>
                        [<Subscriber CGN Port Range TLV>]
                        <Error Information TLV>

The return/error codes are carried in the Error Information TLV. They are listed as follows:

0: Success.

1: Failure. Malformed message received.

2: TLV-Unknown. One or more of the TLVs was not understood.

3001: Pool-Mismatch. The corresponding address pool cannot be found.

3002: Pool-Full. The address pool is fully allocated, and no address segment is available.

3003: Subnet-Mismatch. The address pool subnet cannot be found.

3004: Subnet-Conflict. Subnets in the address pool have been classified into other clients.

4001: Update-Fail-No-Res. The forwarding table fails to be delivered because the forwarding resources are insufficient.

4002: QoS-Update-Success. The QoS policy takes effect.

4003: QoS-Update-Sq-Fail. Failed to process the queue in the QoS policy.

4004: QoS-Update-CAR-Fail. Processing of the CAR in the QoS policy fails.

4005: Statistic-Fail-No-Res. Statistics processing failed due to insufficient statistics resources.

6.3. Event Message

The Event message is used to report subscriber session traffic statistics and detection information. The format of the Event message is as follows:

<Event Message> ::= <Common Header>
                        [<Subscriber Traffic Statistics Report TLV>]
                        [<Subscriber Detection Result Report TLV>]

6.4. Report Message

The Report message is used to report board and interface status on a UP. The format of the Report message is as follows:

<Report Message> ::= <Common Header>
                        [<Board Status TLVs>]

[<Interface Status TLVs>]

6.5. CGN Messages

This document defines the following resource allocation messages:

Type	Message Name	TLV that is carried
200	Addr_Allocation_Req	Address Allocation Request
201	Addr_Allocation_Ack	Address Allocation Response
202	Addr_Renew_Req	Address Renewal Request
203	Addr_Renew_Ack	Address Renewal Response
204	Addr_Release_Req	Address Release Request
205	Addr_Release_Ack	Address Release Response

Table 3: Resource Allocation Messages

6.5.1. Addr_Allocation_Req Message

The Addr_Allocation_Req message is used to request CGN address allocation. The format of the Addr_Allocation_Req message is as follows:

<Addr_Allocation_Req Message> ::= <Common Header>
<Address Allocation Request TLV>

6.5.2. Addr_Allocation_Ack Message

The Addr_Allocation_Ack message is a response to an Addr_Allocation_Req message. The format of the Addr_Allocation_Ack message is as follows:

<Addr_Allocation_Ack Message> ::= <Common Header>
<Address Allocation Response TLV>

6.5.3. Addr_Renew_Req Message

The Addr_Renew_Req message is used to request address renewal. The format of the Addr_Renew_Req message is as follows:

<Addr_Renew_Req Message> ::= <Common Header>
<Address Renewal Request TLV>

6.5.4. Addr_Renew_Ack Message

The Addr_Renew_Ack message is a response to an Addr_Renew_Req message. The format of the Addr_Renew_Req message is as follows:

<Addr_Renew_Ack Message> ::= <Common Header>

<Address Renewal Response TLV>

6.5.5. Addr_Release_Req Message

The Addr_Release_Req message is used to request address release. The format of the Addr_Release_Req message is as follows:

```
<Addr_Release_Req Message> ::= <Common Header>
                               <Address Release Request TLV>
```

6.5.6. Addr_Release_Ack Message

The Addr_Release_Ack message is a response to an Addr_Release_Req message. The format of the Addr_Release_Ack message is as follows:

```
<Addr_Release_Ack Message> ::= <Common Header>
                               <Address Release Response TLV>
```

6.6. Vendor Message

The Vendor message, in conjunction with the Vendor TLV and Vendor sub-TLV, can be used by vendors to extend S-CUSP. The Message-Type is 11. If the receiver does not recognize the message, an Error message will be returned to the sender.

The format of the Vendor message is as follows:

```
<Vendor Message> ::= <Common Header>
                     <Vendor TLV>
                     [<any other TLVs as specified by the vendor>]
```

6.7. Error Message

The Error message is defined to return some critical error information to the sender. If a receiver does not support the type of the received message, it MUST return an Error message to the sender.

The format of the Error message is as below:

```
<Error Message> ::= <Common Header>
                    <Error Information TLV>
```

7. S-CUSP TLVs and Sub-TLVs

This section specifies the following:

* The format of the TLVs that appear in S-CUSP messages,

* The format of the sub-TLVs that appear within the values of some TLVs, and

* The format of some basic data fields that appear within TLVs or sub-TLVs.

See Section 8 for a list of all defined TLVs and sub-TLVs.

7.1. Common TLV Header

S-CUSP messages consist of the common header specified in Section 6.1 followed by TLVs formatted as specified in this section.

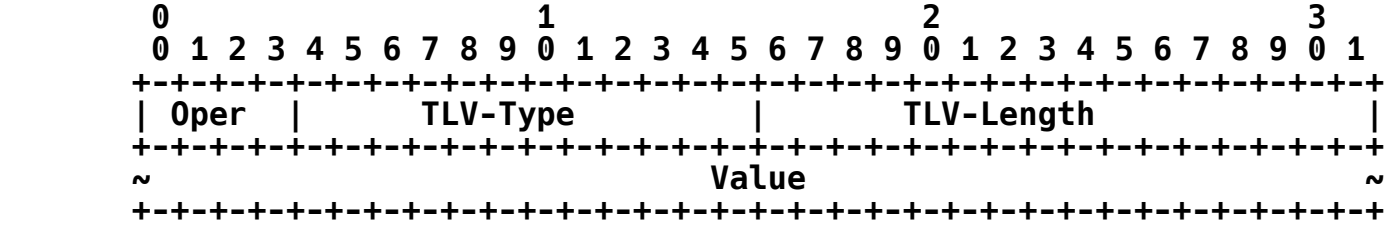


Figure 32: Common TLV Header

Oper (4 bits): For Message-Types that specify an operation on a data set, the Oper field is interpreted as Update, Delete, or Reserved as specified in Section 8.3. For all other Message-Types, the Oper field MUST be sent as zero and ignored on receipt.

TLV-Type (12 bits): The type of a TLV. TLV-Type specifies the interpretation and format of the Value field of the TLV. See Section 8.2.

TLV-Length (2 bytes): The length of the Value portion of the TLV in bytes as an unsigned integer.

Value (variable length): This is the portion of the TLV whose size is given by TLV-Length. It consists of fields, frequently using one of the basic data field types (see Section 7.2) and sub-TLVs (see Section 7.3).

7.2. Basic Data Fields

This section specifies the binary format of several standard basic data fields that are used within other data structures in this specification.

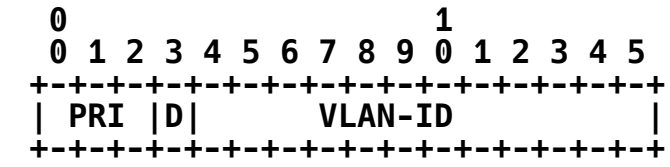
STRING: 0 to 255 octets. Will be encoded as a sub-TLV (see Section 7.3) to provide the length. The use of this data type in S-CUSP is to provide convenient labels for use by network operators in configuring and debugging their networks and interpreting S-CUSP messages. Subscribers will not normally see these labels. They are normally interpreted as ASCII [RFC20].

MAC-Addr: 6 octets. Ethernet MAC address [RFC7042].

IPv4-Address: 8 octets. 4 octets of the IPv4 address value followed by a 4-octet address mask in the format XXX.XXX.XXX.XXX.

IPv6-Address: 20 octets. 16 octets of the IPv6 address followed by a 4-octet integer n in the range of 0 to 128, which gives the address mask as the one's complement of 2**(128-n) - 1.

VLAN ID: 2 octets. As follows [802.1Q]:



PRI: Priority. Default value 7.

D: Drop Eligibility Indicator (DEI). Default value 0.

VLAN-ID: Unsigned integer in the range 1-4094. (0 and 4095 are not valid VLAN IDs [802.1Q].)

7.3. Sub-TLV Format and Sub-TLVs

In some cases, the Value portion of a TLV, as specified in Section 7.1, can contain one or more sub-TLVs formatted as follows:

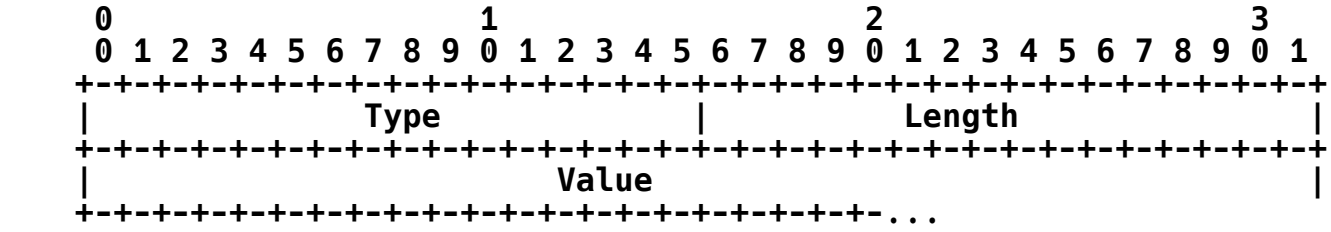


Figure 33: Sub-TLV Header

Type (2 bytes): The type of a sub-TLV. The Type field specifies the interpretation and format of the Value field of the TLV. Sub-TLV type values have the same meaning regardless of the TLV type of the TLV within which the sub-TLV occurs. See Section 8.4.

Length (2 bytes): The length of the Value portion of the sub-TLV in bytes as an unsigned integer.

Value (variable length): This is the Value portion of the sub-TLV whose size is given by Length.

The sub-TLVs currently specified are defined in the following subsections.

7.3.1. Name Sub-TLVs

This document defines the following name sub-TLVs that are used to carry the name of the corresponding object. The length of each of these sub-TLVs is variable from 1 to 255 octets. The value is of type STRING padded with zero octets to a length in octets that is an integer multiple of 4.

Type	Sub-TLV Name	Meaning
1	VRF-Name	The name of a VRF

2	Ingress-QoS-Profile	The name of an ingress QoS profile
3	Egress-QoS-Profile	The name of an egress QoS profile
4	User-ACL-Policy	The name of an ACL policy
5	Multicast-ProfileV4	The name of an IPv4 multicast profile
6	Multicast-ProfileV6	The name of an IPv6 multicast profile
9	NAT-Instance	The name of a NAT instance
10	Pool-Name	The name of an address pool

Table 4: Name Sub-TLVs

7.3.2. Ingress-CAR Sub-TLV

The Ingress-CAR sub-TLV indicates the authorized upstream Committed Access Rate (CAR) parameters. The sub-TLV type of the Ingress-CAR sub-TLV is 7. The sub-TLV length is 16. The format is as shown in Figure 34.

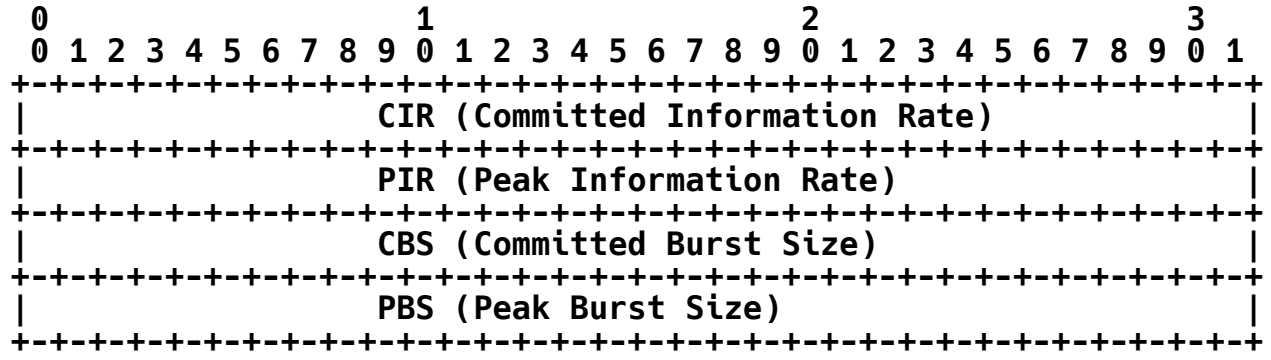


Figure 34: Ingress-CAR Sub-TLV

Where:

CIR (4 bytes): Guaranteed rate in bits/second.

PIR (4 bytes): Burst rate in bits/second.

CBS (4 bytes): The token bucket in bytes.

PBS (4 bytes): Burst token bucket in bytes.

These fields are unsigned integers. More details about CIR, PIR, CBS, and PBS can be found in [RFC2698].

7.3.3. Egress-CAR Sub-TLV

The Egress-CAR sub-TLV indicates the authorized downstream Committed

Access Rate (CAR) parameters. The sub-TLV type of the Egress-CAR sub-TLV is 8. Its sub-TLV length is 16 octets. The format of the value part is as defined below.

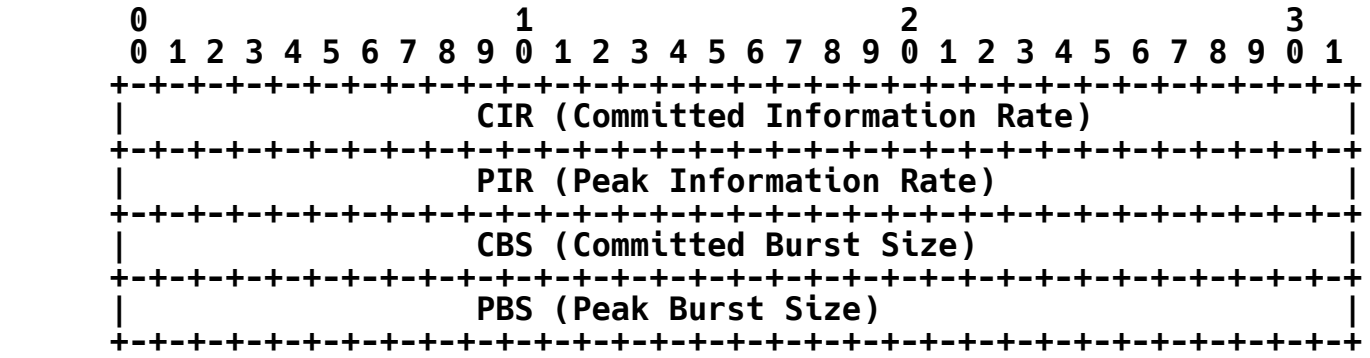


Figure 35: Egress-CAR Sub-TLV

Where:

CIR (4 bytes): Guaranteed rate in bits/second.

PIR (4 bytes): Burst rate in bits/second.

CBS (4 bytes): The token bucket in bytes.

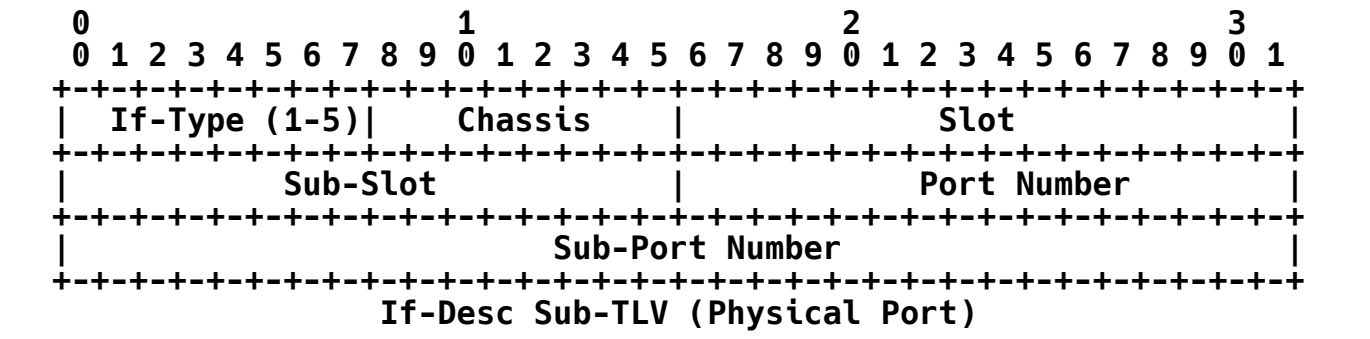
PBS (4 bytes): Burst token bucket in bytes.

These fields are unsigned integers. More details about CIR, PIR, CBS, and PBS can be found in [RFC2698].

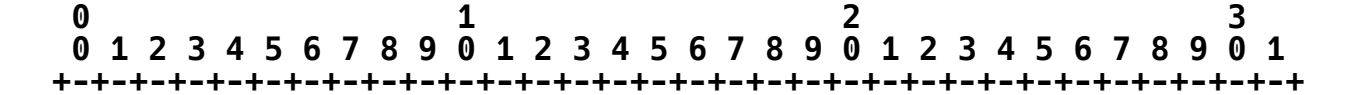
7.3.4. If-Desc Sub-TLV

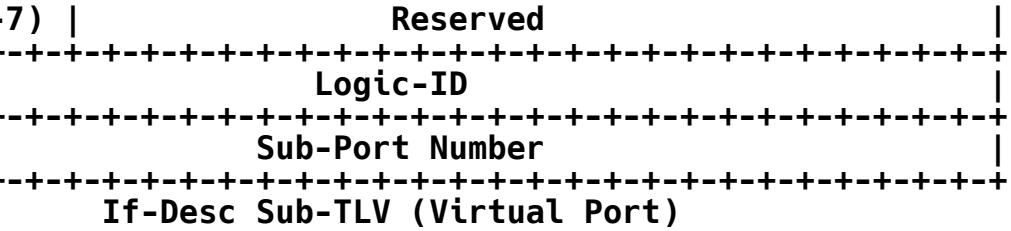
The If-Desc sub-TLV is defined to designate an interface. It is an optional sub-TLV that may be carried in those TLVs that have an If-Index or Out-If-Index field. The If-Desc sub-TLV is used as a locally unique identifier within a BNG.

The sub-TLV type is 11. The sub-TLV length is 12 octets. The format depends on the If-Type (Section 8.6). The format of the value part is as follows:



If-Desc Sub-TLV (Physical Port)





If-Desc Sub-TLV (Virtual Port)

Figure 36: If-Desc Sub-TLV Formats

Where:

If-Type: 8 bits in length. The value of this field indicates the type of an interface. The If-Type values defined in this document are listed in Section 8.6.

Chassis (8 bits): Identifies the chassis that the interface belongs to.

Slot (16 bits): Identifies the slot that the interface belongs to.

Sub-Slot (16 bits): Identifies the sub-slot the interface belongs to.

Port Number (16 bits): An identifier of a physical port/interface (e.g., If-Type: 1-5). It is locally significant within the slot/sub-slot.

Sub-Port Number (32 bits): An identifier of the sub-port. Locally significant within its "parent" port (physical or virtual).

Logic-ID (32 bits): An identifier of a virtual interface (e.g., If-Type: 6-7).

7.3.5. IPv6 Address List Sub-TLV

The IPv6 Address List sub-TLV is used to convey one or more IPv6 addresses. It is carried in the IPv6 Subscriber TLV. The sub-TLV type is 12. The sub-TLV length is variable.

The format of the value part of the IPv6 Address List sub-TLV is as follows:

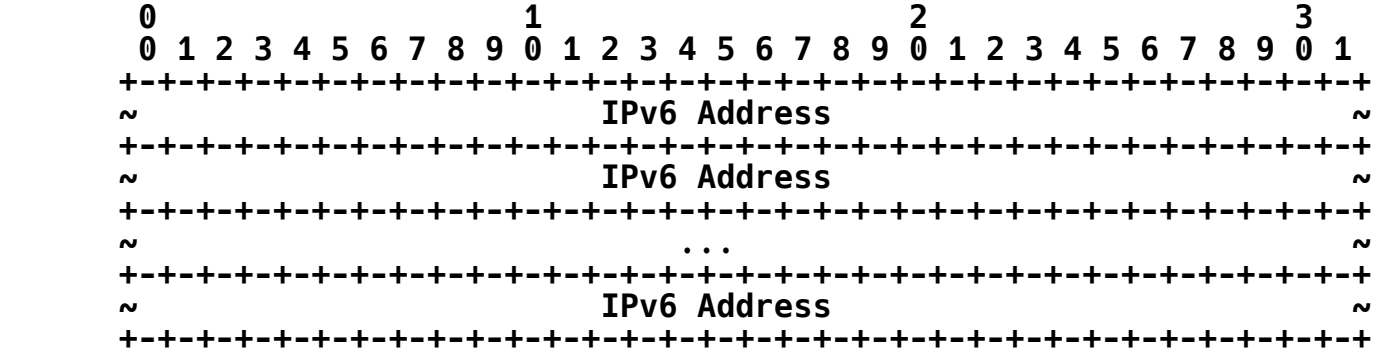


Figure 37: IPv6 Address List Sub-TLV

Where:

IPv6 Address (IPv6-Address): Each IP Address is of type IP-Address and carries an IPv6 address and length.

7.3.6. Vendor Sub-TLV

The Vendor sub-TLV is intended to be used inside the Value portion of the Vendor TLV (Section 7.13). It provides a Sub-Type that effectively extends the sub-TLV type in the sub-TLV header and provides for versioning of Vendor sub-TLVs.

The value part of the Vendor sub-TLV is formatted as follows:

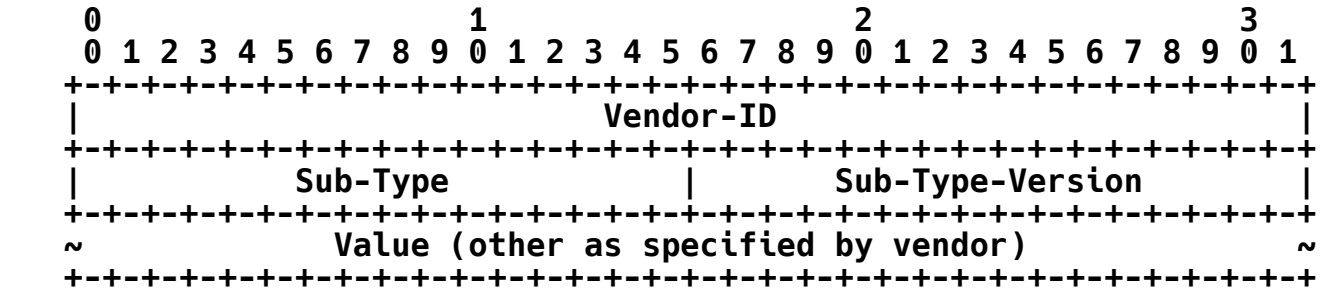


Figure 38: Vendor Sub-TLV

Where:

Sub-TLV type: 13.

Sub-TLV length: Variable.

Vendor-ID (4 bytes): Vendor ID as defined in RADIUS [RFC2865].

Sub-Type (2 bytes): Used by the vendor to distinguish multiple different sub-TLVs.

Sub-Type-Version (2 bytes): Used by the vendor to distinguish different versions of a vendor-defined sub-TLV Sub-Type.

Value: As specified by the vendor.

Since vendor code will be handling the sub-TLV after the Vendor-ID field is recognized, the remainder of the sub-TLV can be organized however the vendor wants. But it desirable for a vendor to be able to define multiple different Vendor sub-TLVs and to keep track of different versions of its vendor-defined sub-TLVs. Thus, it is RECOMMENDED that the vendor assign a Sub-Type value for each of that vendor's sub-TLVs that is different from other Sub-Type values that vendor has used. Also, when modifying a vendor-defined sub-TLV in a way potentially incompatible with a previous definition, the vendor SHOULD increase the value it is using in the Sub-Type-Version field.

7.4. Hello TLV

The Hello TLV is defined to be carried in the Hello message for version and capabilities negotiation. It indicates the S-CUSP sub-version and capabilities supported. The format of the value part of the Hello TLV is as follows:

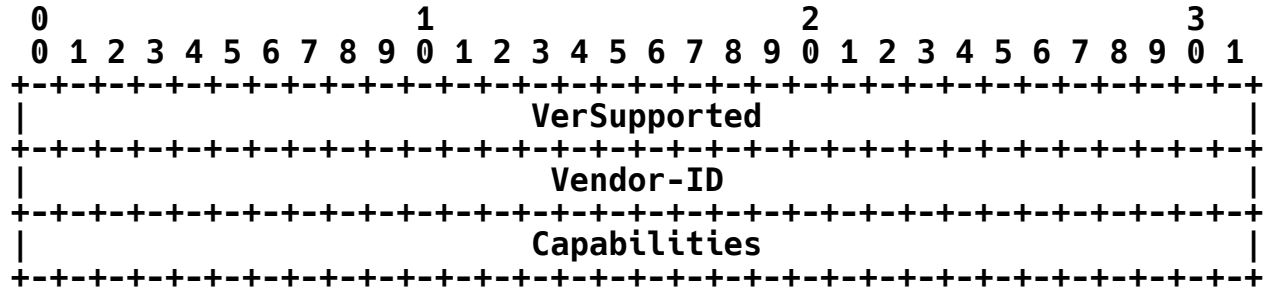


Figure 39: Hello TLV

Where:

TLV type: 100.

TLV length: 12 octets.

VerSupported: 32 bits in length. It is a bit map of the Sub-Versions of S-CUSP that the sender supports. This document specifies Sub-Version zero of Major Version 1, that is, Version 1.0. The VerSupported field MUST be nonzero. The VerSupported bits are numbered from 0 as the most significant bit. Bit 0 indicates support of Sub-Version zero, bit 1 indicates support of Sub-Version one, etc.

Vendor-ID: 4 bytes in length. Vendor ID, as defined in RADIUS [RFC2865].

Capabilities: 32 bits in length. Flags that indicate the support of particular capabilities by the sender of the Hello. No capabilities are defined in this document, so implementations of the version specified herein will set this field to zero. The Capabilities field of the Hello TLV MUST be checked before any other TLVs in the Hello because capabilities defined in the future might extend existing TLVs or permit new TLVs.

After the exchange of Hello messages, the CP and UP each perform a logical AND of the Sub-Version supported by the CP and the UP and separately perform a logical AND of the Capabilities field for the CP and the UP.

If the result of the AND of the Sub-Versions supported is zero, then no session can be established, and the connection is torn down. If the result of the AND of the Sub-Versions supported is nonzero, then the session uses the highest Sub-Version supported by both the CP and UP.

For example, if one side supports Sub-Versions 1, 3, 4, and 5

(VerSupported = 0x5C000000) and the other side supports 2, 3, and 4 (VerSupported = 0x38000000), then 3 and 4 are the Sub-Versions in common, and 4 is the highest Sub-Version supported by both sides. So Sub-Version 4 is used for the session that has been negotiated.

The result of the logical AND of the Capabilities bits will show what additional capabilities both sides support. If this result is zero, there are no such capabilities, so none can be used during the session. If this result is nonzero, it shows the additional capabilities that can be used during the session. The CP and the UP MUST NOT use a capability unless both advertise support.

7.5. Keepalive TLV

The Keepalive TLV is carried in the Hello message. It provides timing information for this feature. The format of the value part of the Keepalive TLV is as follows:

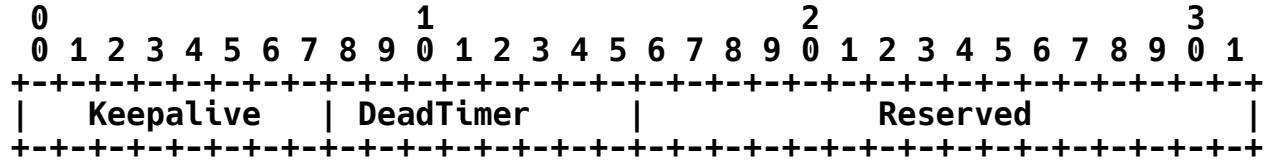


Figure 40: Keepalive TLV

Where:

TLV type: 102.

TLV length: 4 octets.

Keepalive (8 bits): Indicates the maximum interval (in seconds) between two consecutive S-CUSP messages sent by the sender of the message containing this TLV as an unsigned integer. The minimum value for the Keepalive field is 1 second. When set to 0, once the session is established, no further Keepalive messages are sent to the remote peer. A RECOMMENDED value for the Keepalive frequency is 30 seconds.

DeadTimer (8 bits in length): Specifies the amount of time as an unsigned integer number of seconds, after the expiration of which, the S-CUSP peer can declare the session with the sender of the Hello message to be down if no S-CUSP message has been received. The DeadTimer SHOULD be set to 0 and MUST be ignored if the Keepalive is set to 0. A RECOMMENDED value for the DeadTimer is 4 times the value of the Keepalive.

Reserved: The Reserved bits MUST be sent as zero and ignored on receipt.

7.6. Error Information TLV

The Error Information TLV is a common TLV that can be used in many responses (e.g., Update_Response message) and ACK messages (e.g., Addr_Allocation_Ack message). It is used to convey the information

about an error in the received S-CUSP message. The format of the value part of the Error Information TLV is as follows:

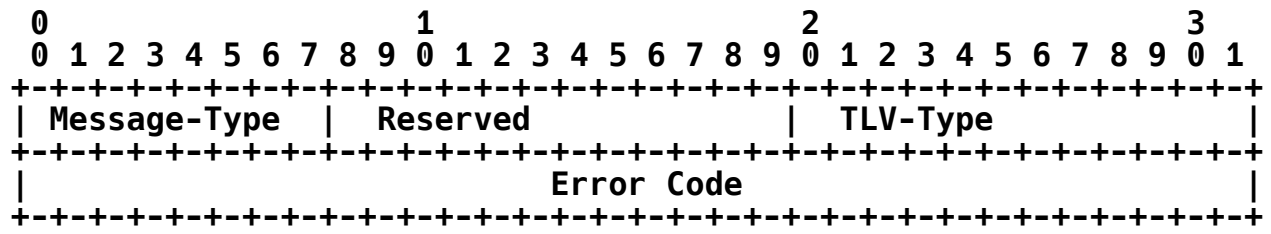


Figure 41: Error Information TLV

Where:

TLV type: 101.

TLV length: 8 octets.

Message-Type (1 byte): This parameter is the message type of the message containing an error.

Reserved (1 byte): MUST be sent as zero and ignored on receipt.

TLV-Type (2 bytes): Indicates which TLV caused the error.

Error Code: 4 bytes in length. Indicate the specific Error Code (see Section 8.5).

7.7. BAS Function TLV

The BAS Function TLV is used by a CP to control the access mode, authentication methods, and other related functions of an interface on a UP.

The format of the BAS Function TLV value part is as follows:

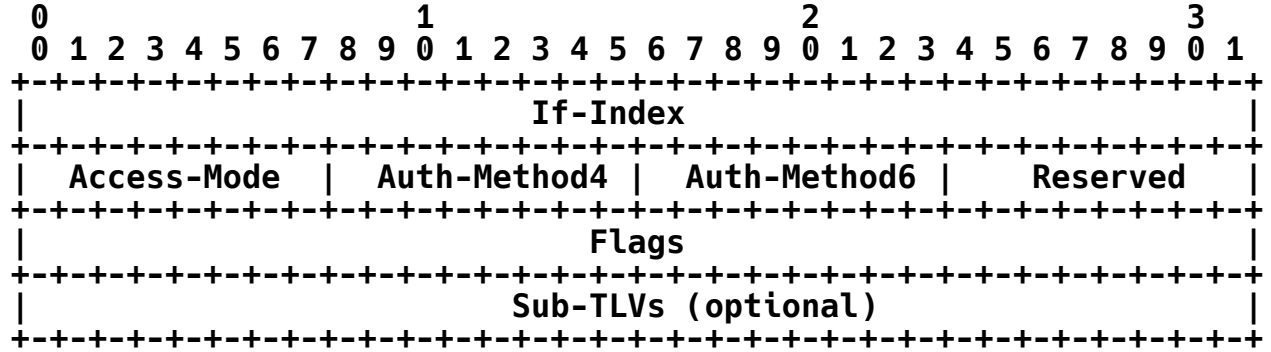


Figure 42: BAS Function TLV

Where:

TLV type: 1.

TLV length: Variable.

If-Index: 4 bytes in length, a unique identifier of an interface of a BNG.

Access-Mode: 1 byte in length. It indicates the access mode of the interface. The defined values are listed in Section 8.7.

Auth-Method4: 1 byte in length. It indicates the authentication on this interface for the IPv4 scenario. This field is defined as a bitmap. The bits defined in this document are listed in Section 8.8. Other bits are reserved and MUST be sent as zero and ignored on receipt.

Auth-Method6: 1 byte in length. It indicates the authentication on this interface for the IPv6 scenario. This field is defined as a bitmap. The bits defined in this document are listed in Section 8.8. Other bits are reserved and MUST be sent as zero and ignored on receipt.

Sub-TLVs: The IF-Desc sub-TLV can be carried.

If-Desc sub-TLV: Carries the interface information.

Flags: The Flags field is defined as follows:

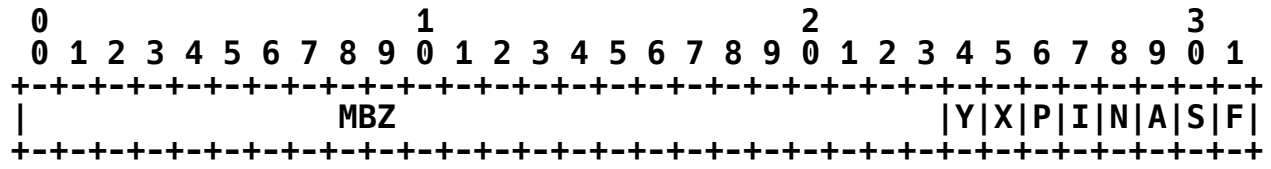


Figure 43: Interface Flags

Where:

F (IPv4 Trigger) bit: Indicates whether IPv4 packets can trigger a subscriber to go online.

1: Enabled.

0: Disabled.

S (IPv6 Trigger) bit: Indicates whether IPv6 packets can trigger a subscriber to go online.

1: Enabled.

0: Disabled.

A (ARP Trigger) bit: Indicates whether ARP packets can trigger a subscriber to go online.

1: Enabled.

0: Disabled.

N (ND Trigger) bit: Indicates whether ND packets can trigger a subscriber to go online.

1: Enabled.

0: Disabled.

I (IPoE-Flow-Check): Used for UP detection.

1: Enable traffic detection.

0: Disable traffic detection.

P (PPP-Flow-Check) bit: Used for UP detection.

1: Enable traffic detection.

0: Disable traffic detection.

X (ARP-Proxy) bit: Indicates whether ARP proxy is enabled on the interface.

1: The interface is enabled with ARP proxy and can process ARP requests across different network ports and VLANs.

0: The ARP proxy is not enabled on the interface and only the ARP requests of the same network port and VLAN are processed.

Y (ND-Proxy) bit: Indicates whether ND proxy is enabled on the interface.

1: The interface is enabled with ND proxy and can process ND requests across different network ports and VLANs.

0: The ND proxy is not enabled on the interface and only the ND requests of the same network port and VLAN are processed.

MBZ: Reserved bits that MUST be sent as zero and ignored on receipt.

7.8. Routing TLVs

Typically, after an S-CUSP session is established between a UP and a CP, the CP will allocate one or more blocks of IP addresses to the UP. Those IP addresses will be allocated to subscribers who will dial-up (as defined in Section 4.3.1) to the UP. To make sure that other nodes within the network learn how to reach those IP addresses, the CP needs to install one or more routes that can reach those IP addresses on the UP and notify the UP to advertise the routes to the network.

The Routing TLVs are used by a CP to notify a UP of the updates to network routing information. They can be carried in the Update_Request message and Sync_Data message.

7.8.1. IPv4 Routing TLV

The IPv4 Routing TLV is used to carry information related to IPv4 network routing.

The format of the TLV value part is as below:

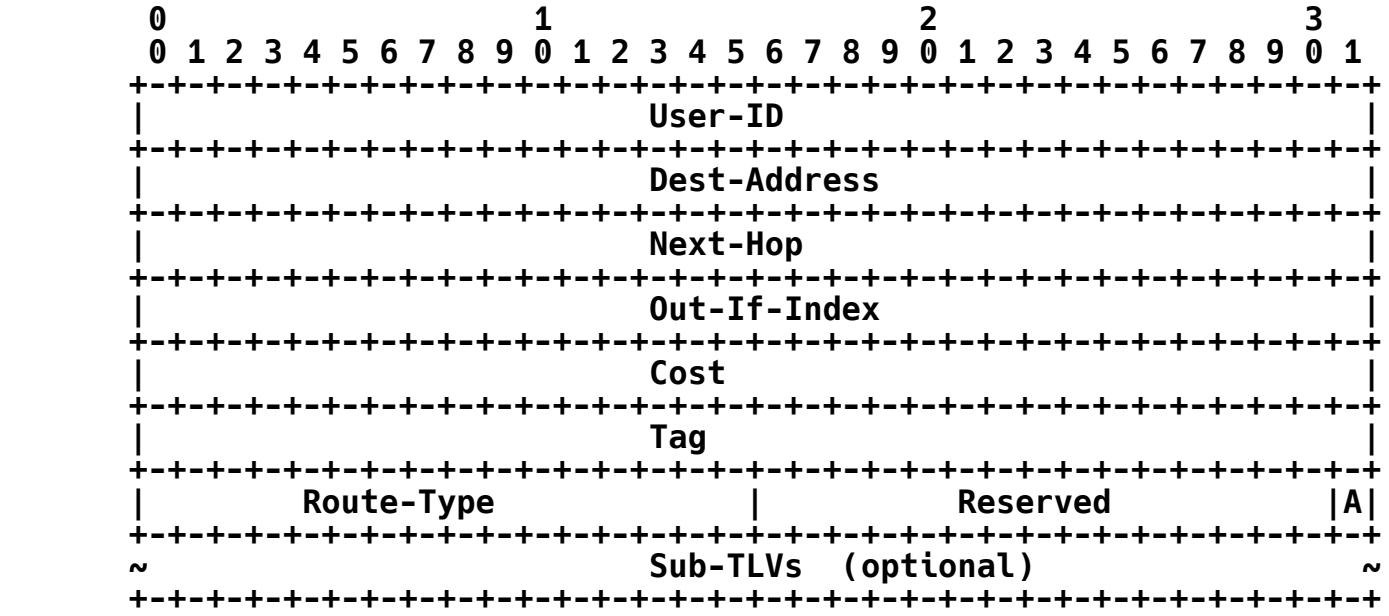


Figure 44: IPv4 Routing TLV

Where:

TLV type: 7.

TLV length: Variable.

User-ID: 4 bytes in length. This field carries the user identifier. It is filled with all Fs when a non-user route is delivered to the UP.

Dest-Address (IPv4-Address type): Identifies the destination address.

Next-Hop (IPv4-Address type): Identifies the next-hop address.

Out-If-Index (4 bytes): Indicates the interface index.

Cost (4 bytes): The cost value of the route.

Tag (4 bytes): The tag value of the route.

Route-Type (2 bytes): The value of this field indicates the route type. The values defined in this document are listed in Section 8.9.

Advertise-Flag: 1 bit shown as "A" in the figure above (Figure 44). Indicates whether the UP should advertise the

route. The following flag values are defined:

0: Not advertised.

1: Advertised.

Sub-TLVs: The VRF-Name and/or If-Desc sub-TLVs can be carried.

VRF-Name sub-TLV: Indicates which VRF the route belongs to.

If-Desc sub-TLV: Carries the interface information.

Reserved: The Reserved field MUST be sent as zero and ignored on receipt.

7.8.2. IPv6 Routing TLV

The IPv6 Routing TLV is used to carry IPv6 network routing information.

The format of the value part of this TLV is as follows:

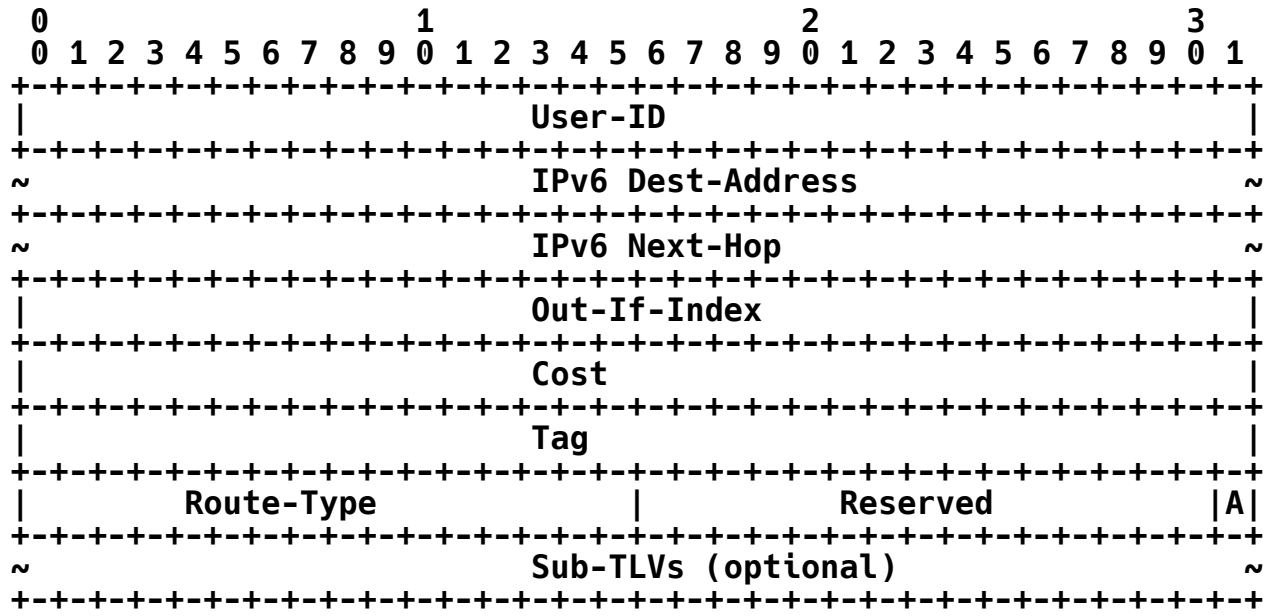


Figure 45: IPv6 Routing TLV

Where:

TLV type: 8.

TLV length: Variable.

User-ID: 4 bytes in length. This field carries the user identifier. This field is filled with all Fs when a non-user route is delivered to the UP.

IPv6 Dest-Address (IPv6-Address type): Identifies the destination address.

IPv6 Next-Hop (IPv6-Address type): Identifies the next-hop address.

Out-If-Index (4 bytes): Indicates the interface index.

Cost (4 bytes): This is the cost value of the route.

Tag (4 bytes): The tag value of the route.

Route-Type (2 bytes): The value of this field indicates the route type. The values defined in this document are listed in Section 8.9.

Advertise-Flag: 1 bit shown as "A" in the figure above (Figure 45). Indicates whether the UP should advertise the route. The following flags are defined:

0: Not advertised.

1: Advertised.

Sub-TLVs: The If-Desc and VRF-Name sub-TLVs can be carried.

VRF-Name sub-TLV: Indicates the VRF to which the subscriber belongs.

If-Desc sub-TLV: Carries the interface information.

Reserved: The Reserved field MUST be sent as zero and ignored on receipt.

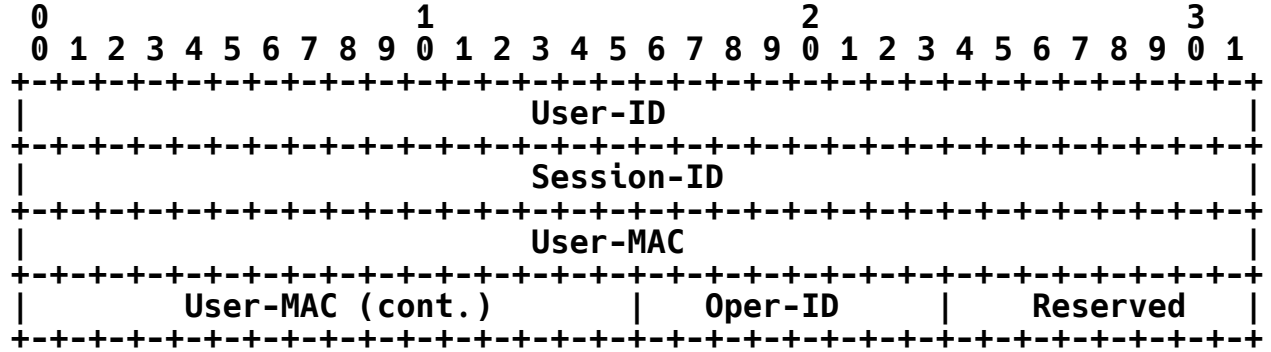
7.9. Subscriber TLVs

The Subscriber TLVs are defined for a CP to send the basic information about a user to a UP.

7.9.1. Basic Subscriber TLV

The Basic Subscriber TLV is used to carry the common information for all kinds of access subscribers. It is carried in an Update_Request message.

The format of the Basic Subscriber TLV value part is as follows:



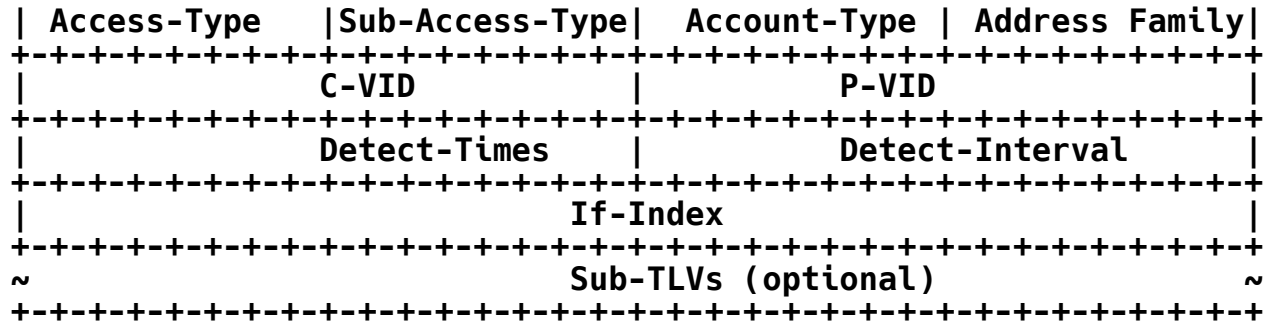


Figure 46: Basic Subscriber TLV

Where:

TLV type: 2.

TLV length: Variable.

User-ID (4 bytes): The identifier of a subscriber.

Session-ID (4 bytes): Session ID of a PPPoE subscriber. The value zero identifies a non-PPPoE subscriber.

User-MAC (MAC-Addr type): The MAC address of a subscriber.

Oper-ID (1 byte): Indicates the ID of an operation performed by a user. This field is carried in the response from the UP.

Reserved (1 byte): MUST be sent as zero and ignored on receipt.

Access-Type (1 byte): Indicates the type of subscriber access. Values defined in this document are listed in Section 8.10.

Sub-Access-Type (1 byte): Indicates whether PPP termination or PPP relay is used.

0: Reserved.

1: PPP Relay (for LAC).

2: PPP termination (for LNS).

Account-Type (1 byte): Indicates whether traffic statistics are collected independently.

0: Collects statistics on IPv4 and IPv6 traffic of terminals independently.

1: Collects statistics on IPv4 and IPv6 traffic of terminals.

Address Family (1 byte): The type of IP address.

1: IPv4.

2: IPv6.

3: Dual stack.

C-VID (VLAN-ID): Indicates the inner VLAN ID. The value 0 indicates that the VLAN ID is invalid. The default value of PRI is 7, the value of DEI is 0, and the value of VID is 1-4094. The PRI value can also be obtained by parsing terminal packets.

P-VID (VLAN-ID): Indicates the outer VLAN ID. The value 0 indicates that the VLAN ID is invalid. The format is the same as that for C-VID.

Detect-Times (2 bytes): Number of detection timeout times. The value 0 indicates that no detection is performed.

Detect-Interval (2 bytes): Detection interval in seconds.

If-Index (4 bytes): Interface index.

Sub-TLVs: The VRF-Name sub-TLV and If-Desc sub-TLV can be carried.

VRF-Name sub-TLV: Indicates the VRF to which the subscriber belongs.

If-Desc sub-TLV: Carries the interface information.

Reserved: The Reserved field MUST be sent as zero and ignored on receipt.

7.9.2. PPP Subscriber TLV

The PPP Subscriber TLV is defined to carry PPP information of a user from a CP to a UP. It will be carried in an Update_Request message when PPPoE or L2TP access is used.

The format of the TLV value part is as follows:

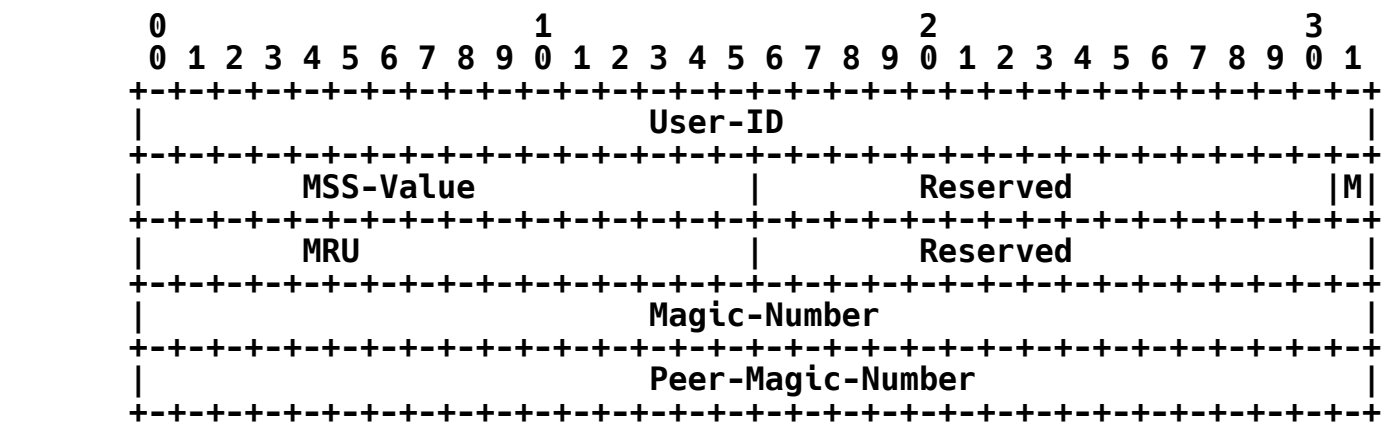


Figure 47: PPP Subscriber TLV

Where:

TLV type: 3.

TLV length: 12 octets.

User-ID (4 bytes): The identifier of a subscriber.

MSS-Value (2 bytes): Indicates the MSS value (in bytes).

MSS-Enable (M) (1 bit): Indicates whether the MSS is enabled.

0: Disabled.

1: Enabled.

MRU (2 bytes): PPPoE local MRU (in bytes).

Magic-Number (4 bytes): Local magic number in PPP negotiation packets.

Peer-Magic-Number (4 bytes): Remote peer magic number.

Reserved: The Reserved fields MUST be sent as zero and ignored on receipt.

7.9.3. IPv4 Subscriber TLV

The IPv4 Subscriber TLV is defined to carry IPv4-related information for a BNG user. It will be carried in an Update_Request message when IPv4 IPoE or PPPoE access is used.

The format of the TLV value part is as follows:

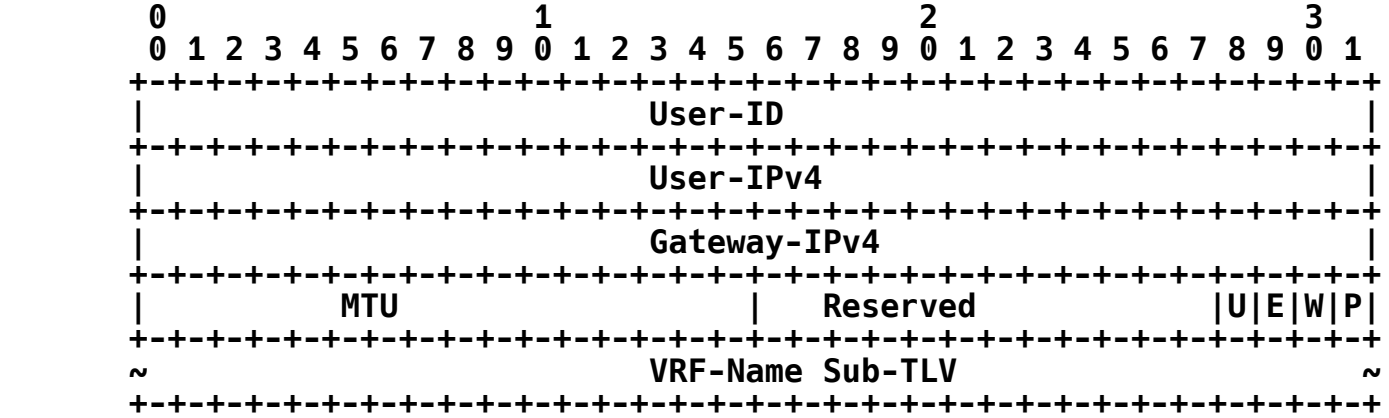


Figure 48: IPv4 Subscriber TLV

Where:

TLV type: 4.

TLV length: Variable.

User-ID (4 bytes): The identifier of a subscriber.

User-IPv4 (IPv4-Address): The IPv4 address of the subscriber.

Gateway-IPv4 (IPv4-Address): The gateway address of the subscriber.

Portal-Force (P) (1 bit): Push advertisement.

0: Off.

1: On.

Web-Force (W) (1 bit): Push IPv4 Web.

0: Off.

1: On.

Echo-Enable (E) (1 bit): UP returns ARP Req or PPP Echo.

0: Off.

1: On.

IPv4-URPF (U) (1 bit): User Unicast Reverse Path Forwarding (URPF) flag.

0: Off.

1: On.

MTU (2 bytes): MTU value. The default value is 1500.

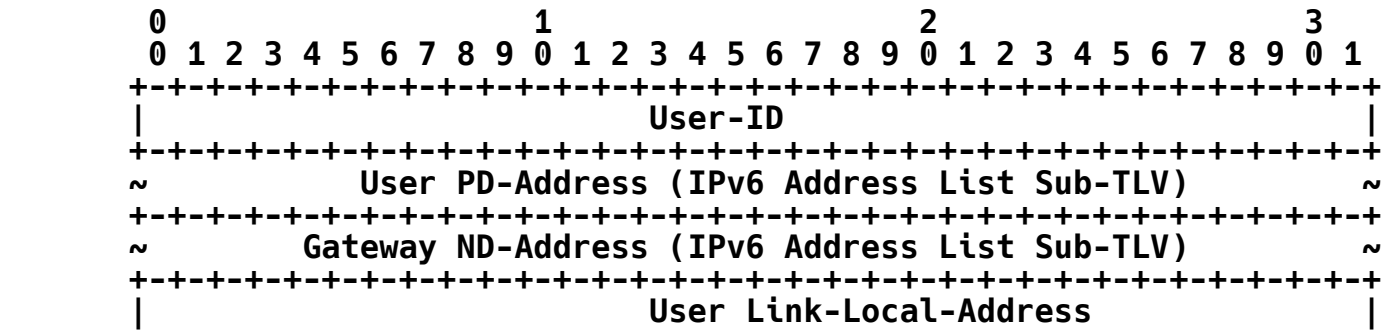
VRF-Name Sub-TLV: Indicates the subscriber belongs to which VRF.

Reserved: The Reserved field MUST be sent as zero and ignored on receipt.

7.9.4. IPv6 Subscriber TLV

The IPv6 Subscriber TLV is defined to carry IPv6-related information for a BNG user. It will be carried in an Update_Request message when IPv6 IPoE or PPPoE access is used.

The format of the TLV value part is as follows:



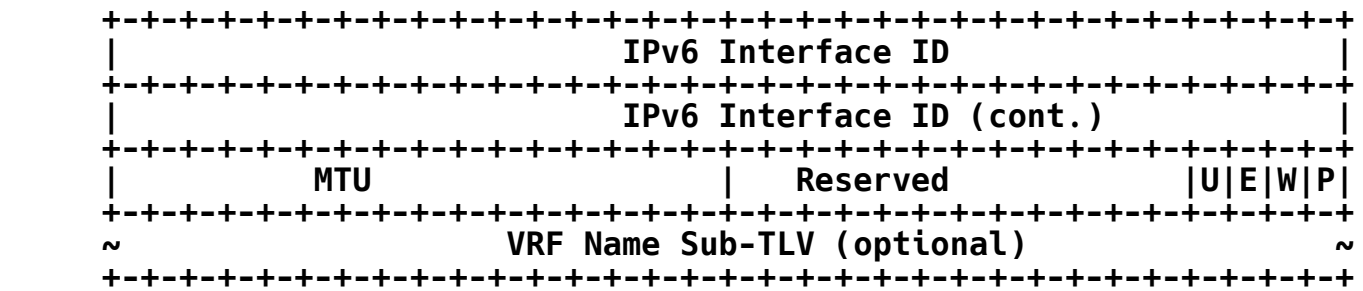


Figure 49: IPv6 Subscriber TLV

Where:

TLV type: 5.

TLV length: Variable.

User-ID (4 bytes): The identifier of a subscriber.

User PD-Addresses (IPv6 Address List): Carries a list of Prefix Delegation (PD) addresses of the subscriber.

User ND-Addresses (IPv6 Address List): Carries a list of Neighbor Discovery (ND) addresses of the subscriber.

User Link-Local-Address (IPv6-Address): The link-local address of the subscriber.

IPv6 Interface ID (8 bytes): The identifier of an IPv6 interface.

Portal-Force 1 bit (P): Push advertisement.

0: Off.

1: On.

Web-Force 1 bit (W): Push IPv6 Web.

0: Off.

1: On.

Echo-Enable 1 bit (E): The UP returns ARP Req or PPP Echo.

0: Off.

1: On.

IPv6-URPF 1 bit (U): User Reverse Path Forwarding (URPF) flag.

0: Off.

1: On.

MTU (2 bytes): The MTU value. The default value is 1500.

VRF-Name Sub-TLV: Indicates the VRF to which the subscriber belongs.

Reserved: The Reserved field MUST be sent as zero and ignored on receipt.

7.9.5. IPv4 Static Subscriber Detect TLV

The IPv4 Static Subscriber Detect TLV is defined to carry IPv4-related information for a static access subscriber. It will be carried in an Update_Request message when IPv4 static access on a UP needs to be enabled.

The format of the TLV value part is as follows:

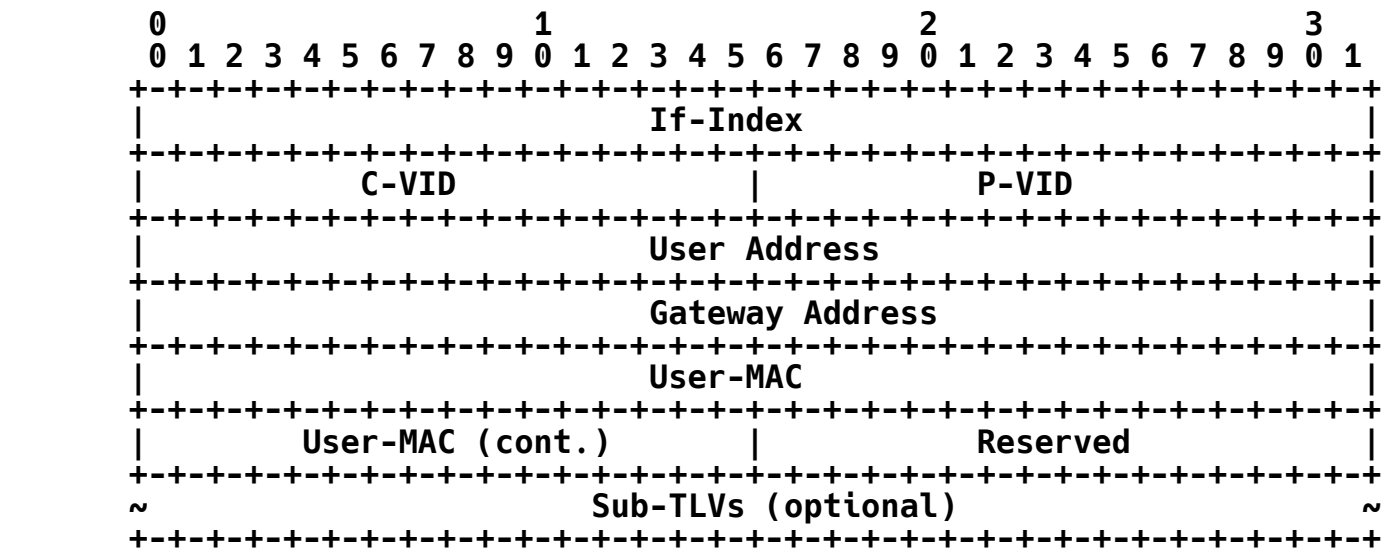


Figure 50: IPv4 Static Subscriber TLV

Where:

TLV type: 9.

TLV length: Variable.

If-Index (4 bytes): The interface index of the interface from which the subscriber will dial-up.

C-VID (VLAN-ID): Indicates the inner VLAN ID. The value 0 indicates that the VLAN ID is invalid. A valid value is 1-4094.

P-VID (VLAN-ID): Indicates the outer VLAN ID. The value 0 indicates that the VLAN ID is invalid. The format is the same as that of the C-VID. A valid value is 1-4094.

User Address (IPv4-Addr): The user's IPv4 address.

Gateway Address (IPv4-Addr): The gateway's IPv4 address.

User-MAC (MAC-Addr type): The MAC address of the subscriber.

Sub-TLVs: The VRF-Name and If-Desc sub-TLVs may be carried.

VRF-Name sub-TLV: Indicates the VRF to which the subscriber belongs.

If-Desc sub-TLV: Carries the interface information.

Reserved: The Reserved field MUST be sent as zero and ignored on receipt.

7.9.6. IPv6 Static Subscriber Detect TLV

The IPv6 Static Subscriber Detect TLV is defined to carry IPv6-related information for a static access subscriber. It will be carried in an Update_Request message when needed to enable IPv6 static subscriber detection on a UP.

The format of the TLV value part is as follows:

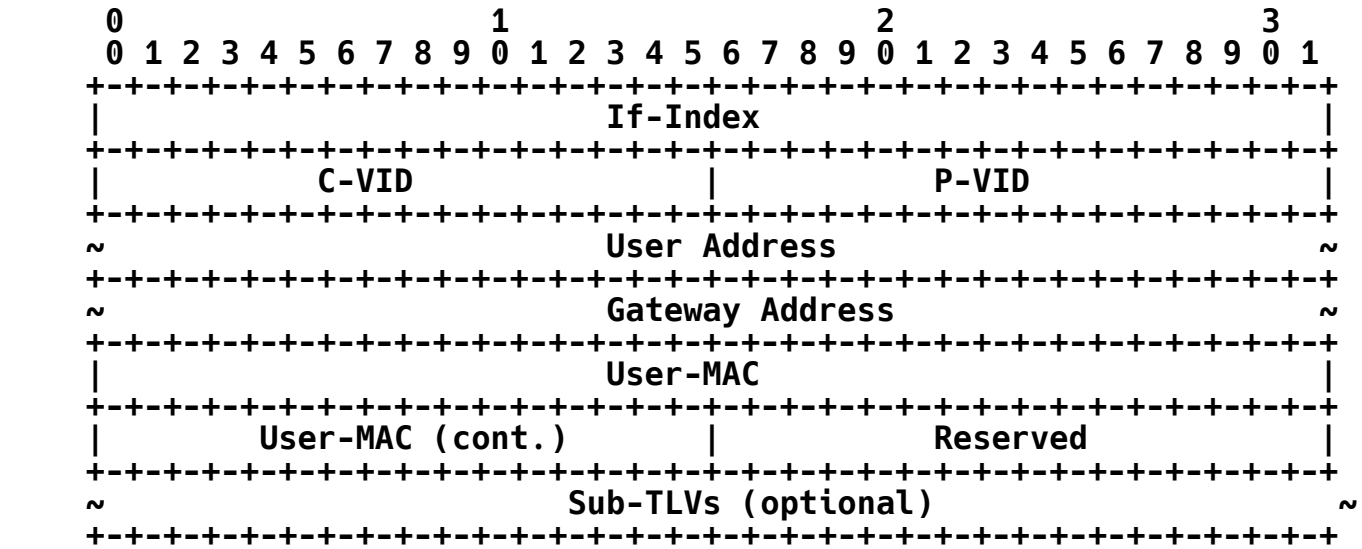


Figure 51: IPv6 Static Subscriber Detect TLV

Where:

TLV type: 10.

TLV length: Variable.

If-Index (4 bytes): The interface index of the interface from which the subscriber will dial-up.

C-VID (VLAN-ID): Indicates the inner VLAN ID. The value 0 indicates that the VLAN ID is invalid. A valid value is 1-4094.

P-VID (VLAN-ID): Indicates the outer VLAN ID. The value 0

indicates that the VLAN ID is invalid. The format is the same as that the of C-VID. A valid value is 1-4094.

User Address (IPv6-Address type): The subscriber's IPv6 address.

Gateway Address (IPv6-Address type): The gateway's IPv6 Address.

User-MAC (MAC-Addr type): The MAC address of the subscriber.

Sub-TLVs: VRF-Name and If-Desc sub-TLVs may be carried

VRF-Name sub-TLV: Indicates the VRF to which the subscriber belongs.

If-Desc sub-TLV: Carries the interface information.

Reserved: The Reserved field MUST be sent as zero and ignored on receipt.

7.9.7. L2TP-LAC Subscriber TLV

The L2TP-LAC Subscriber TLV is defined to carry the related information for an L2TP LAC access subscriber. It will be carried in an Update_Request message when L2TP LAC access is used.

The format of the TLV value part is as follows:

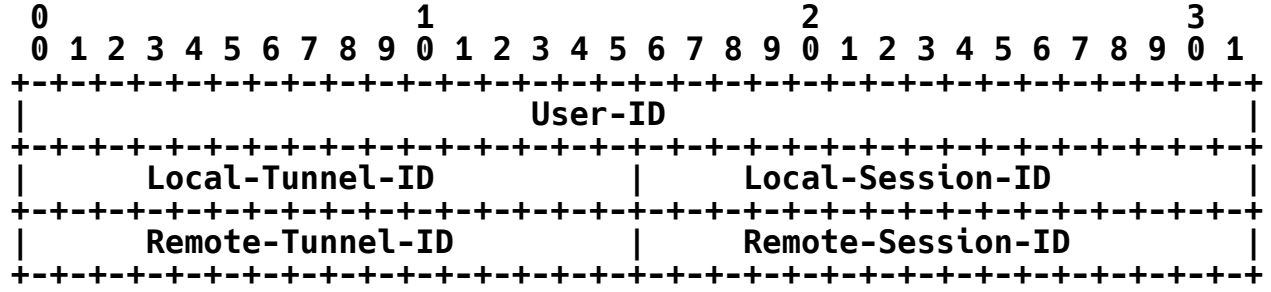


Figure 52: L2TP-LAC Subscriber TLV

Where:

TLV type: 11.

TLV length: 12 octets.

User-ID (4 bytes): The identifier of a user/subscriber.

Local-Tunnel-ID (2 bytes): The local ID of the L2TP tunnel.

Local-Session-ID (2 bytes): The local session ID with the L2TP tunnel.

Remote-Tunnel-ID (2 bytes): The identifier of the L2TP tunnel at the remote endpoint.

Remote-Session-ID (2 bytes): The session ID of the L2TP tunnel at

the remote endpoint.

7.9.8. L2TP-LNS Subscriber TLV

The L2TP-LNS Subscriber TLV is defined to carry the related information for a L2TP LNS access subscriber. It will be carried in an Update_Request message when L2TP LNS access is used.

The format of the TLV value part is as follows:

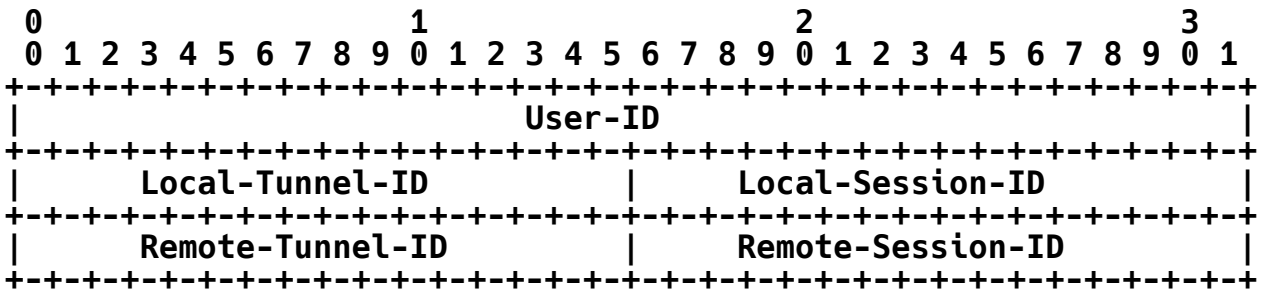


Figure 53: L2TP-LNS Subscriber TLV

Where:

TLV type: 12.

TLV length: 12 octets.

User-ID (4 bytes): The identifier of a user/subscriber.

Local-Tunnel-ID (2 bytes): The local ID of the L2TP tunnel.

Local-Session-ID (2 bytes): The local session ID with the L2TP tunnel.

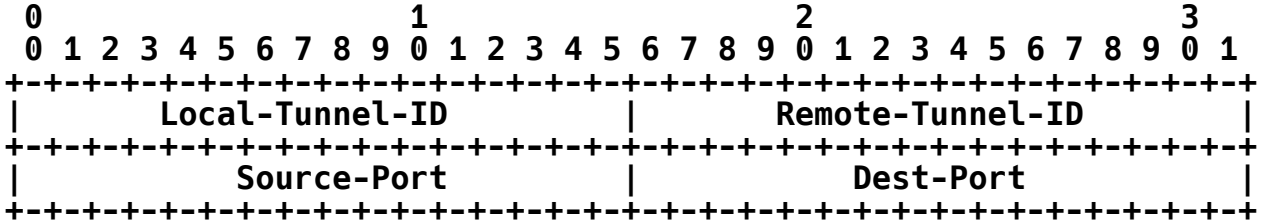
Remote-Tunnel-ID (2 bytes): The identifier of the L2TP tunnel at the remote endpoint.

Remote-Session-ID (2 bytes): The session ID of the L2TP tunnel at the remote endpoint.

7.9.9. L2TP-LAC Tunnel TLV

The L2TP-LAC Tunnel TLV is defined to carry information related to the L2TP LAC tunnel. It will be carried in the Update_Request message when L2TP LAC access is used.

The format of the TLV value part is as follows:



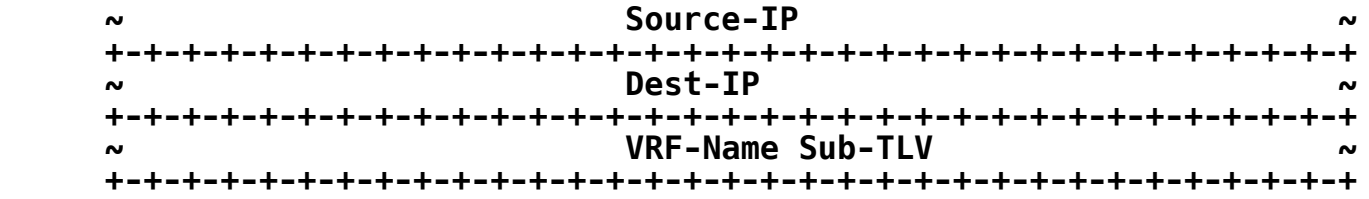


Figure 54: L2TP-LAC Tunnel TLV

Where:

TLV type: 13.

TLV length: Variable.

Local-Tunnel-ID (2 bytes): The local ID of the L2TP tunnel.

Remote-Tunnel-ID (2 bytes): The remote ID of the L2TP tunnel.

Source-Port (2 bytes): The source UDP port number of an L2TP subscriber.

Dest-Port (2 bytes): The destination UDP port number of an L2TP subscriber.

Source-IP (IPv4/v6): The source IP address of the tunnel.

Dest-IP (IPv4/v6): The destination IP address of the tunnel.

VRF-Name Sub-TLV: The VRF name to which the L2TP subscriber tunnel belongs.

7.9.10. L2TP-LNS Tunnel TLV

The L2TP-LNS Tunnel TLV is defined to carry information related to the L2TP LNS tunnel. It will be carried in the Update_Request message when L2TP LNS access is used.

The format of the TLV value part is as follows:

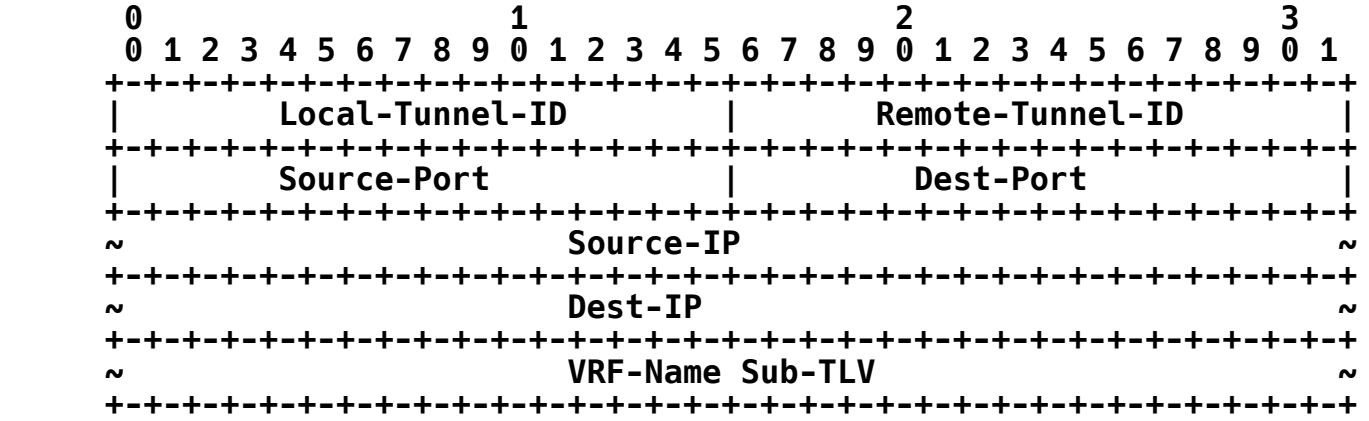


Figure 55: L2TP-LNS Tunnel TLV

Where:

TLV type: 14.

TLV length: Variable.

Local-Tunnel-ID (2 bytes): The local ID of the L2TP tunnel.

Remote-Tunnel-ID (2 bytes): The remote ID of the L2TP tunnel.

Source-Port (2 bytes): The source UDP port number of an L2TP subscriber.

Dest-Port (2 bytes): The destination UDP port number of an L2TP subscriber.

Source-IP (IPv4/v6): The source IP address of the tunnel.

Dest-IP (IPv4/v6): The destination IP address of the tunnel.

VRF-Name Sub-TLV: The VRF name to which the L2TP subscriber tunnel belongs.

7.9.11. Update Response TLV

The Update Response TLV is used to return the operation result of an update request. It is carried in the Update_Response message as a response to the Update_Request message.

The format of the value part of the Update Response TLV is as follows:

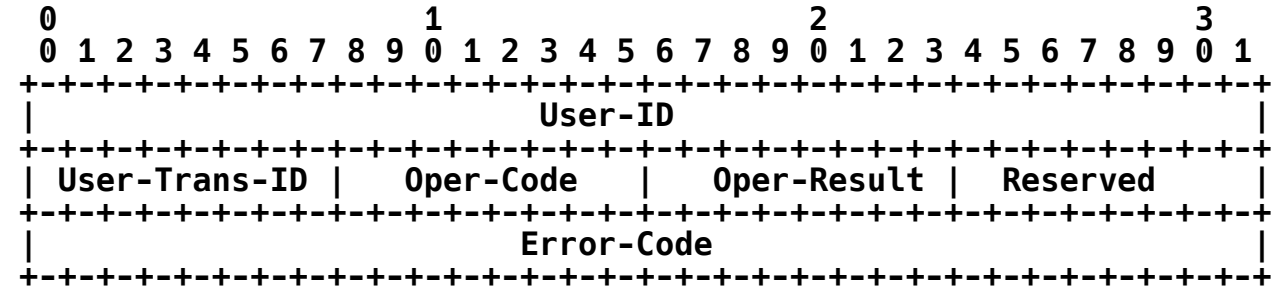


Figure 56: Update Response TLV

Where:

TLV type: 302.

TLV length: 12.

User-ID (4 bytes): A unique identifier of a user/subscriber.

User-Trans-ID (1 byte): In the case of dual-stack access or when modifying a session, User-Trans-ID is used to identify a user operation transaction. It is used by the CP to correlate a response to a specific request.

Oper-Code (1 byte): Operation code.

1: Update.

2: Delete.

Oper-Result (1 byte): Operation Result.

0: Success.

Others: Failure.

Error-Code (4 bytes): Operation failure cause code. For details, see Section 8.5.

Reserved: The Reserved field MUST be sent as zero and ignored on receipt.

7.9.12. Subscriber Policy TLV

The Subscriber Policy TLV is used to carry the policies that will be applied to a subscriber. It is carried in the Update_Request message.

The format of the TLV value part is as follows:

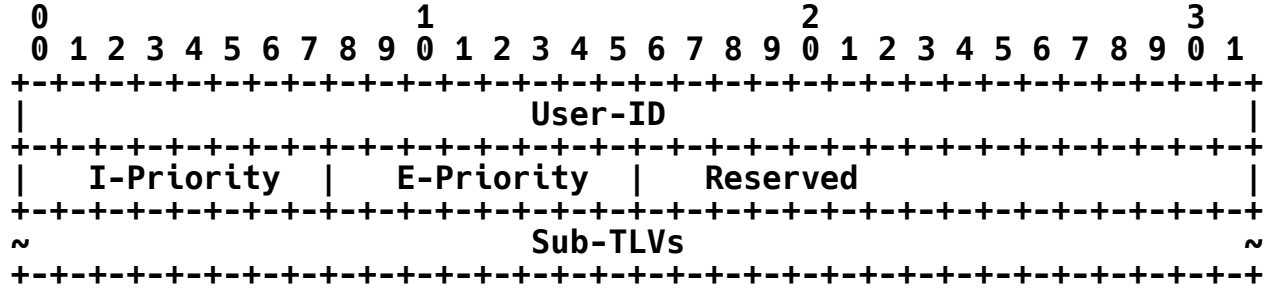


Figure 57: Subscriber Policy TLV

Where:

TLV type: 6.

TLV length: Variable.

User-ID (4 bytes): The identifier of a user/subscriber.

Ingress-Priority (1 byte): Indicates the upstream priority. The value range is 0~7.

Egress-Priority (1 byte): Indicates the downstream priority. The value range is 0~7.

Sub-TLVs: The sub-TLVs that are present can occur in any order.

Ingress-CAR sub-TLV: Upstream CAR.

Egress-CAR sub-TLV: Downstream CAR.

Ingress-QoS-Profile sub-TLV: Indicates the name of the QoS-Profile that is the profile in the upstream direction.

Egress-QoS-Profile sub-TLV: Indicates the name of the QoS-Profile that is the profile in the downstream direction.

User-ACL-Policy sub-TLV: All ACL user policies, including v4ACLIN, v4ACLOUT, v6ACLIN, v6ACLOUT, v4WEBACL, v6WEBACL, v4SpecialACL, and v6SpecialACL.

Multicast-Profile4 sub-TLV: IPv4 multicast policy name.

Multicast-Profile6 sub-TLV: IPv6 multicast policy name.

NAT-Instance sub-TLV: Indicates the instance ID of a NAT user.

Reserved: The Reserved field MUST be sent as zero and ignored on receipt.

7.9.13. Subscriber CGN Port Range TLV

The Subscriber CGN Port Range TLV is used to carry the NAT public address and port range. It will be carried in the Update_Response message when CGN is used.

The format of the value part of this TLV is as follows:

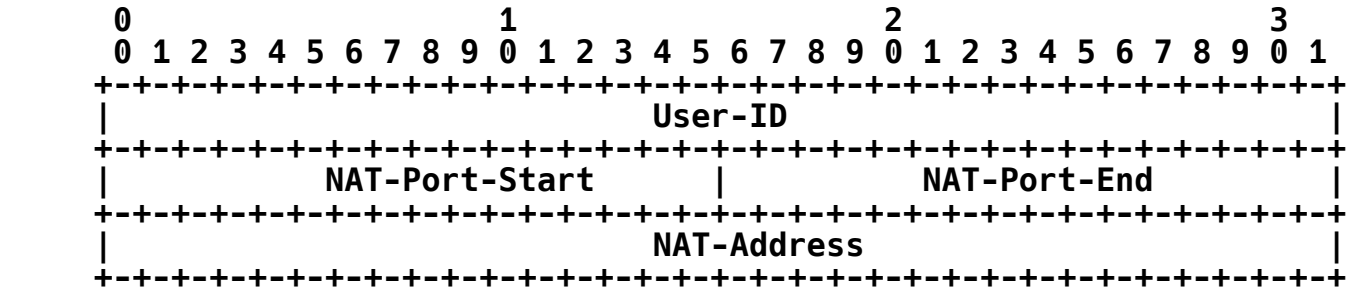


Figure 58: Subscriber CGN Port Range TLV

Where:

TLV type: 15.

TLV length: 12 octets.

User-ID (4 bytes): The identifier of a user/subscriber.

NAT-Port-Start (2 bytes): The start port number.

NAT-Port-End (2 bytes): The end port number.

NAT-Address (4 bytes): The NAT public network address.

7.10. Device Status TLVs

The TLVs in this section are for reporting interface and board-level information from the UP to the CP.

7.10.1. Interface Status TLV

The Interface Status TLV is used to carry the status information of an interface on a UP. It is carried in a Report message.

The format of the value part of this TLV is as follows:

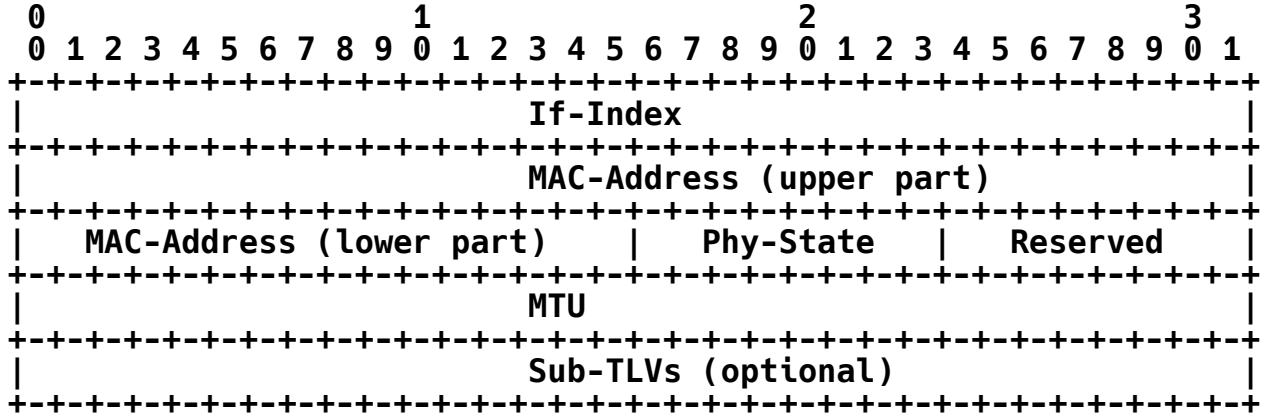


Figure 59: Interface Status TLV

Where:

TLV type: 200.

TLV length: Variable.

If-Index (4 bytes): Indicates the interface index.

MAC-Address (MAC-Addr type): Interface MAC address.

Phy-State (1 byte): Physical status of the interface.

- 0: Down.
- 1: Up.

MTU (4 bytes): Interface MTU value.

Sub-TLVs: The If-Desc and VRF-Name sub-TLVs can be carried.

Reserved: The Reserved field MUST be sent as zero and ignored on receipt.

7.10.2. Board Status TLV

The Board Status TLV is used to carry the status information of a board on an UP. It is carried in a Report message.

The format of the value part of the Board Status TLV is as follows:

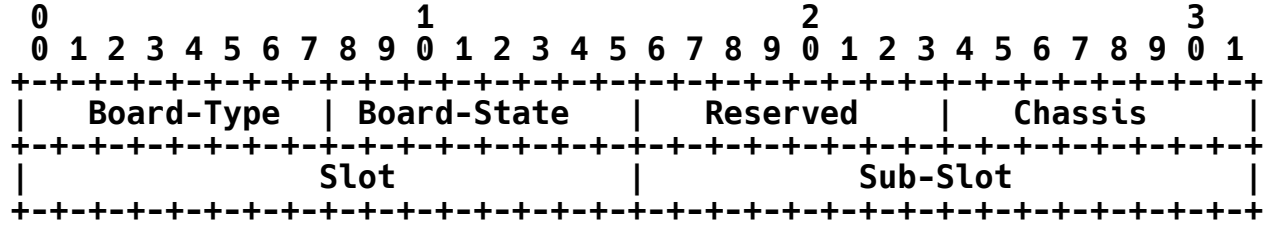


Figure 60: Board Status TLV

Where:

TLV type: 201.

TLV length: 8 octets.

Chassis (1 byte): The chassis number of the board.

Slot (16 bits): The slot number of the board.

Sub-Slot (16 bits): The sub-slot number of the board.

Board-Type (1 byte): The type of board used.

1: CGN Service Process Unit (SPU) board.

2: Line Process Unit (LPU) board.

Board-State (1 byte): Indicates whether there are issues with the board.

0: Normal.

1: Abnormal.

Reserved: The Reserved field MUST be sent as zero and ignored on receipt.

7.11. CGN TLVs

7.11.1. Address Allocation Request TLV

The Address Allocation Request TLV is used to request address allocation from the CP. A Pool-Name sub-TLV is carried to indicate from which address pool to allocate addresses. The Address Allocation Request TLV is carried in the Addr_Allocation_Req message.

The format of the value part of this TLV is as follows:

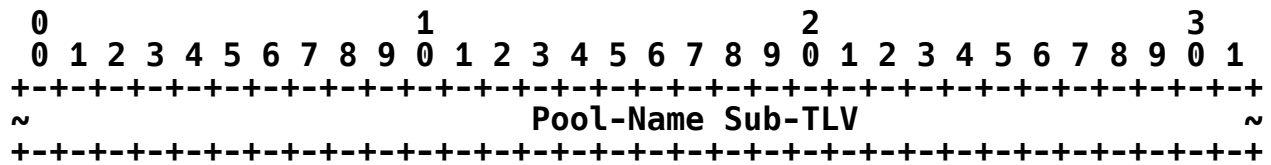


Figure 61: Address Allocation Request TLV

Where:

TLV type: 400.

TLV length: Variable.

Pool-Name sub-TLV: Indicates from which address pool to allocate address.

7.11.2. Address Allocation Response TLV

The Address Allocation Response TLV is used to return the address allocation result; it is carried in the Addr_Allocation_Ack message.

The value part of the Address Allocation Response TLV is formatted as follows:

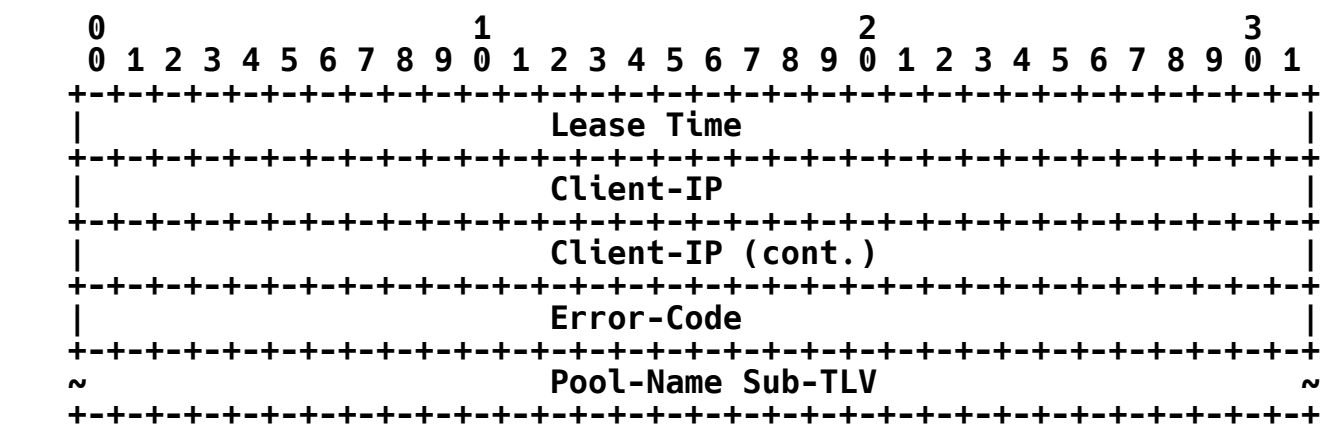


Figure 62: Address Allocation Response TLV

Where:

TLV type: 401.

TLV length: Variable.

Lease Time (4 bytes): Duration of address lease in seconds.

Client-IP (IPv4-Address type): The allocated IPv4 address and mask.

Error-Code (4 bytes): Indicates success or an error.

0: Success.

1: Failure.

3001: Pool-Mismatch. The corresponding address pool cannot be found.

3002: Pool-Full. The address pool is fully allocated, and no address segment is available.

Pool-Name sub-TLV: Indicates from which address pool the address is allocated.

7.11.3. Address Renewal Request TLV

The Address Renewal Request TLV is used to request address renewal from the CP. It is carried in the Addr_Renew_Req message.

The format of this TLV value is as follows:

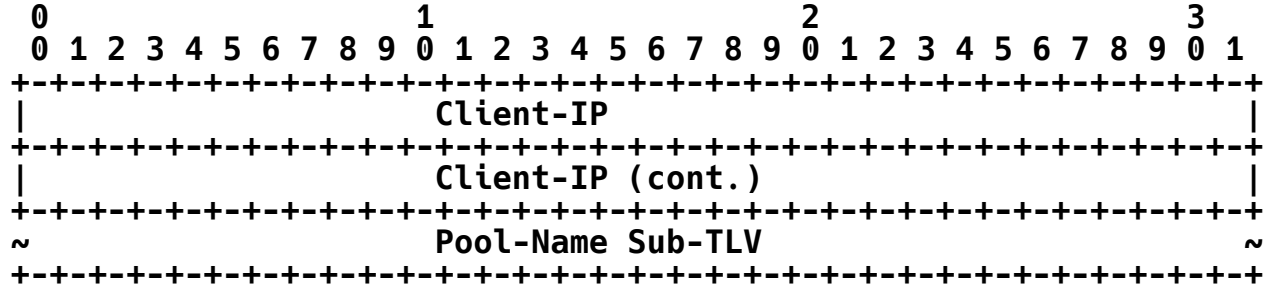


Figure 63: Address Renewal Request TLV

Where:

TLV type: 402.

TLV length: Variable.

Client-IP (IPv4-Address type): The IPv4 address and mask to be renewed.

Pool-Name sub-TLV: Indicates from which address pool to renew the address.

7.11.4. Address Renewal Response TLV

The Address Renewal Response TLV is used to return the address renewal result. It is carried in the Addr_Renew_Ack message.

The format of this TLV value is as follows:

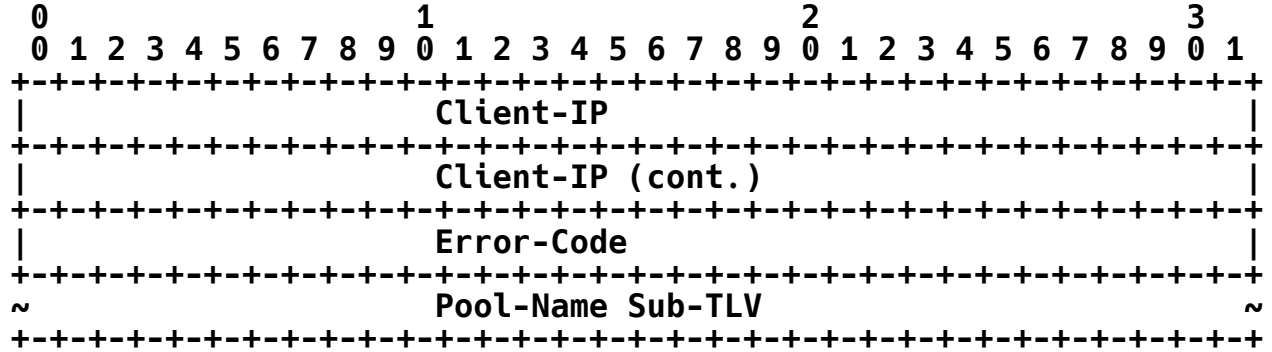


Figure 64: Address Renewal Response TLV

Where:

TLV type: 403.

TLV length: Variable.

Client-IP (IPv4-Address type): The renewed IPv4 address and mask.

Error-Code (4 bytes): Indicates success or an error:

0: Success.

1: Failure.

3001: Pool-Mismatch. The corresponding address pool cannot be found.

3002: Pool-Full. The address pool is fully allocated, and no address segment is available.

3003: Subnet-Mismatch. The address pool subnet cannot be found.

3004: Subnet-Conflict. Subnets in the address pool have been assigned to other clients.

Pool-Name sub-TLV: Indicates from which address pool to renew the address.

7.11.5. Address Release Request TLV

The Address Release Request TLV is used to release an IPv4 address. It is carried in the Addr_Release_Req message.

The value part of this TLV is formatted as follows:

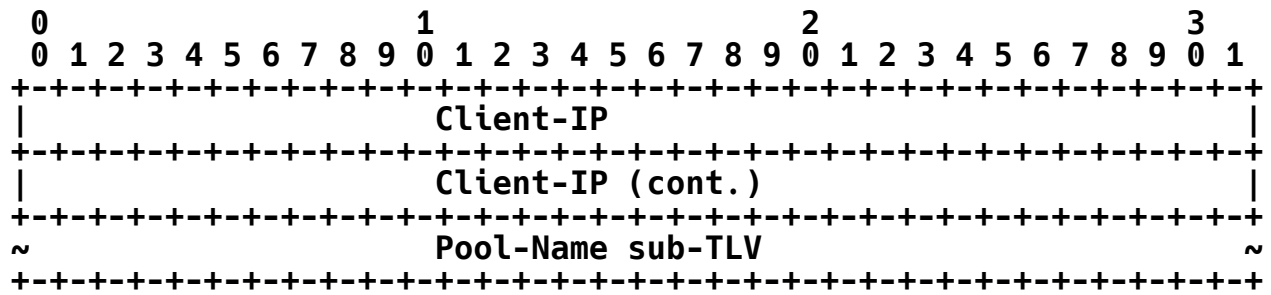


Figure 65: Address Release Request TLV

Where:

TLV type: 404.

TLV length: Variable.

Client-IP (IPv4-Address type): The IPv4 address and mask to be released.

Pool-Name sub-TLV: Indicates from which address pool to release the address.

7.11.6. Address Release Response TLV

The Address Release Response TLV is used to return the address release result. It is carried in the Addr_Release_Ack message.

The format of the value part of this TLV is as follows:

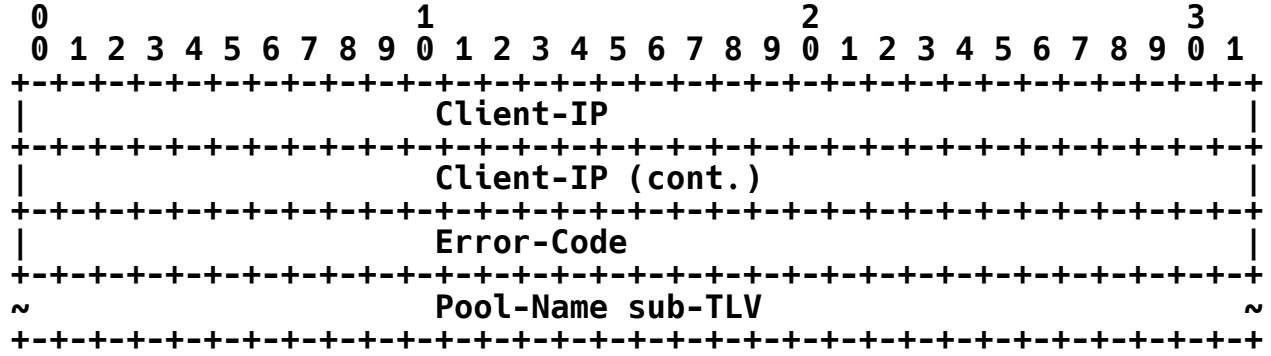


Figure 66: Address Release Response TLV

Where:

TLV type: 405.

TLV length: Variable.

Client-IP (IPv4-Address type): The released IPv4 address and mask.

Error-Code (4 bytes): Indicates success or an error.

- 0: Success. Address release success.
- 1: Failure. Address release failed.
- 3001: Pool-Mismatch. The corresponding address pool cannot be found.
- 3003: Subnet-Mismatch. The address cannot be found.
- 3004: Subnet-Conflict. The address has been allocated to another subscriber.

Pool-Name sub-TLV: Indicates from which address pool to release the address.

7.12. Event TLVs

7.12.1. Subscriber Traffic Statistics TLV

The Subscriber Traffic Statistics TLV is used to return the traffic statistics of a user/subscriber. The format of the value part of this TLV is as follows:

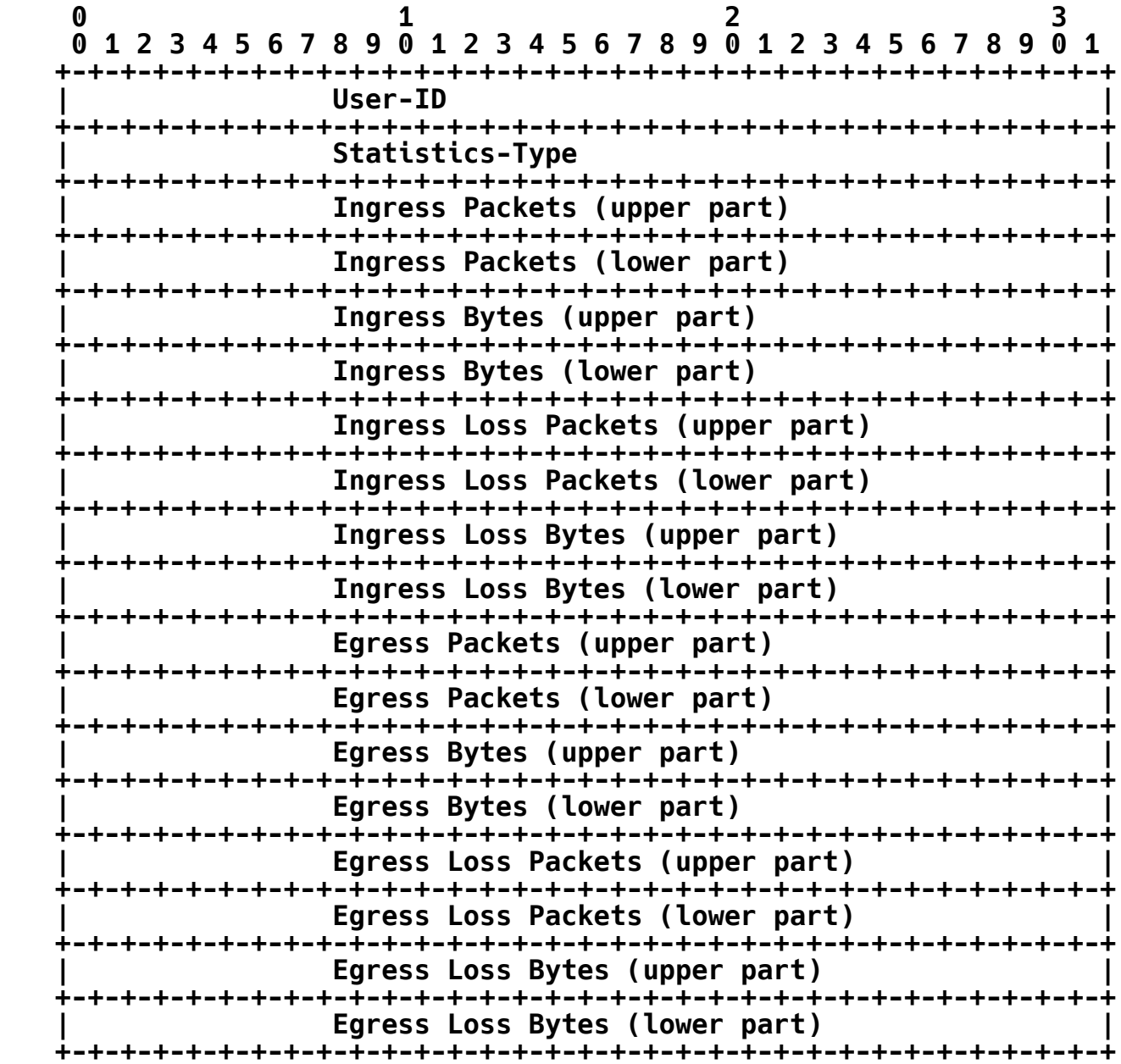


Figure 67: Subscriber Traffic Statistics TLV

Where:

TLV type: 300.

TLV length: 72 octets.

User-ID (4 bytes): The subscriber identifier.

Statistics-Type (4 bytes): Traffic type. It can be one of the

following options:

0: IPv4 traffic.

1: IPv6 traffic.

2: Dual-stack traffic.

Ingress Packets (8 bytes): The number of the packets in the upstream direction.

Ingress Bytes (8 bytes): The bytes of the upstream traffic.

Ingress Loss Packets (8 bytes): The number of the lost packets in the upstream direction.

Ingress Loss Bytes (8 bytes): The bytes of the lost upstream packets.

Egress Packets (8 bytes): The number of the packets in the downstream direction.

Egress Bytes (8 bytes): The bytes of the downstream traffic.

Egress Loss Packets (8 bytes): The number of the lost packets in the downstream direction.

Egress Loss Bytes (8 bytes): The bytes of the lost downstream packets.

7.12.2. Subscriber Detection Result TLV

The Subscriber Detection Result TLV is used to return the detection result of a subscriber. Subscriber detection is a function to detect whether or not a subscriber is online. The result can be used by the CP to determine how to deal with the subscriber session (e.g., delete the session if detection failed).

The format of this TLV value part is as follows:

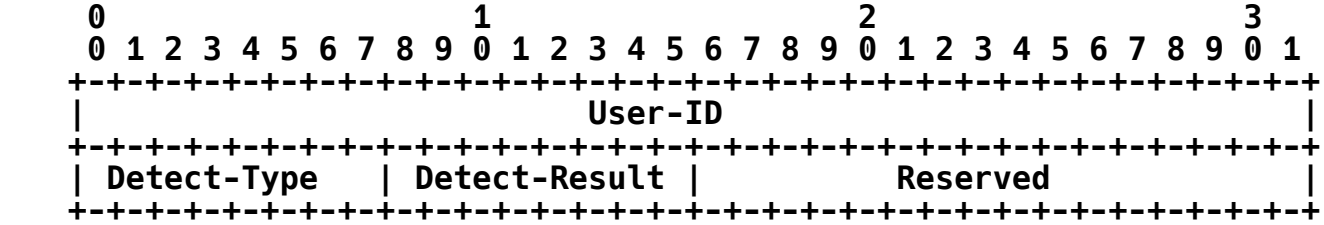


Figure 68: Subscriber Detection Result TLV

Where:

TLV type: 301.

TLV length: 8 octets.

User-ID (4 bytes): The subscriber identifier.

Detect-Type (1 byte): Type of traffic detected.

0: IPv4 detection.

1: IPv6 detection.

2: PPP detection.

Detect-Result (1 byte): Indicates whether the detection was successful.

0: Indicates that the detection is successful.

1: Detection failure. The UP needs to report only when the detection fails.

Reserved: The Reserved field MUST be sent as zero and ignored on receipt.

7.13. Vendor TLV

The Vendor TLV occurs as the first TLV in the Vendor message (Section 6.6). It provides a Sub-Type that effectively extends the message type in the message header, provides for versioning of vendor TLVs, and can accommodate sub-TLVs.

The value part of the Vendor TLV is formatted as follows:

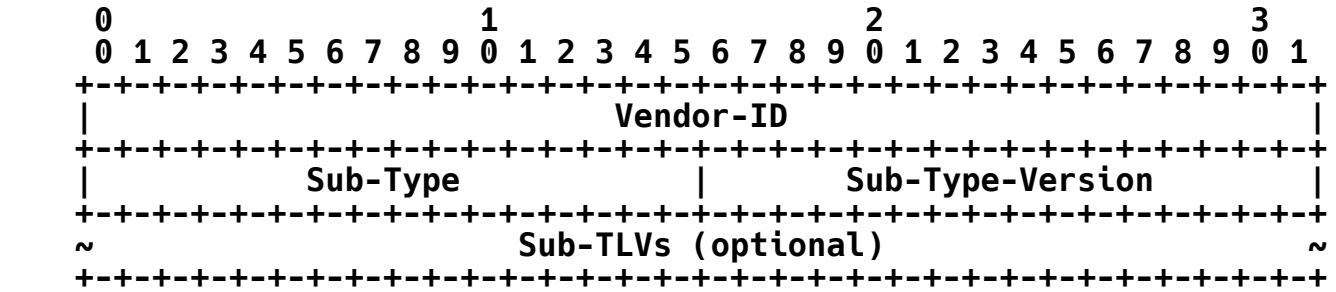


Figure 69: Vendor TLV

Where:

TLV type: 1024.

TLV length: Variable.

Vendor-ID (4 bytes): Vendor ID as defined in RADIUS [RFC2865].

Sub-Type (2 bytes): Used by the vendor to distinguish multiple different vendor messages.

Sub-Type-Version (2 bytes): Used by the vendor to distinguish different versions of a vendor-defined message Sub-Type.

Sub-TLVs (variable): Sub-TLVs as specified by the vendor.

Since vendor code will be handling the TLV after the Vendor-ID field is recognized, the remainder of the TLV values can be organized however the vendor wants. But it is desirable for a vendor to be able to define multiple different vendor messages and to keep track of different versions of its vendor-defined messages. Thus, it is RECOMMENDED that the vendor assign a Sub-Type value for each vendor message that it defines different from other Sub-Type values that vendor has used. Also, when modifying a vendor-defined message in a way potentially incompatible with a previous definition, the vendor SHOULD increase the value it is using in the Sub-Type-Version field.

8. Tables of S-CUSP Codepoints

This section provides tables of the S-CUSP codepoints, particularly message types, TLV types, TLV operation codes, sub-TLV types, and error codes. In most cases, references are provided to relevant sections elsewhere in this document.

8.1. Message Types

Type	Name	Section of This Document
0	Reserved	
1	Hello	6.2.1
2	Keepalive	6.2.2
3	Sync_Request	6.2.3
4	Sync_Begin	6.2.4
5	Sync_Data	6.2.5
6	Sync_End	6.2.6
7	Update_Request	6.2.7
8	Update_Response	6.2.8
9	Report	6.4
10	Event	6.3
11	Vendor	6.6
12	Error	6.7
13-199	Unassigned	
200	Addr_Allocation_Req	6.5.1
201	Addr_Allocation_Ack	6.5.2

202	Addr_Renew_Req	6.5.3
203	Addr_Renew_Ack	6.5.4
204	Addr_Release_Req	6.5.5
205	Addr_Release_Ack	6.5.6
206-254	Unassigned	
255	Reserved	

Table 5: Message Types

8.2. TLV Types

Type	Name	Usage Description
0	Reserved	-
1	BAS Function	Carries the BNG access functions to be enabled or disabled on specified interfaces.
2	Basic Subscriber	Carries the basic information about a BNG subscriber.
3	PPP Subscriber	Carries the PPP information about a BNG subscriber.
4	IPv4 Subscriber	Carries the IPv4 address of a BNG subscriber.
5	IPv6 Subscriber	Carries the IPv6 address of a BNG subscriber.
6	Subscriber Policy	Carries the policy information applied to a BNG subscriber.
7	IPv4 Routing	Carries the IPv4 network routing information.
8	IPv6 Routing	Carries the IPv6 network routing information.
9	IPv4 Static Subscriber Detect	Carries the IPv4 static subscriber detect information.
10	IPv6 Static Subscriber Detect	Carries the IPv6 static subscriber detect information.

11	L2TP-LAC Subscriber	Carries the L2TP LAC subscriber information.
12	L2TP-LNS Subscriber	Carries the L2TP LNS subscriber information.
13	L2TP-LAC Tunnel	Carries the L2TP LAC tunnel subscriber information.
14	L2TP-LNS Tunnel	Carries the L2TP LNS tunnel subscriber information.
15	Subscriber CGN Port Range	Carries the public IPv4 address and related port range of a BNG subscriber.
16-99	Unassigned	-
100	Hello	Used for version and Keepalive timers negotiation.
101	Error Information	Carried in the Ack of the control message. Carries the processing result, success, or error.
102	Keepalive	Carried in the Hello message for Keepalive timers negotiation.
103-199	Unassigned	-
200	Interface Status	Interfaces status reported by the UP including physical interfaces, sub-interfaces, trunk interfaces, and tunnel interfaces.
201	Board Status	Board information reported by the UP including the board type and in-position status.
202-299	Unassigned	-
300	Subscriber Traffic Statistics	User traffic statistics.
301	Subscriber Detection Result	User detection information.
302	Update Response	The processing result of a subscriber session update.
303-299	Unassigned	-
400	Address Allocation	Request address allocation.

	Request	
401	Address Allocation Response	Address allocation response.
402	Address Renewal Request	Request for address lease renewal.
403	Address Renewal Response	Response to a request for extending an IP address lease.
404	Address Release Request	Request to release the address.
405	Address Release Response	Ack of a message releasing an IP address.
406-1023	Unassigned	-
1024	Vendor	As implemented by the vendor.
1039-4095	Unassigned	-

Table 6: TLV Types

8.3. TLV Operation Codes

TLV operation codes appear in the Oper field in the header of some TLVs. See Section 7.1.

Code	Operation
0	Reserved
1	Update
2	Delete
3-15	Unassigned

Table 7: TLV Operation Codes

8.4. Sub-TLV Types

See Section 7.3.

Type	Name	Section of This Document
0	Reserved	
1	VRF Name	7.3.1
2	Ingress-QoS-Profile	7.3.1
3	Egress-QoS-Profile	7.3.1
4	User-ACL-Policy	7.3.1
5	Multicast-ProfileV4	7.3.1
6	Multicast-ProfileV6	7.3.1
7	Ingress-CAR	7.3.2
8	Egress-CAR	7.3.3
9	NAT-Instance	7.3.1
10	Pool-Name	7.3.1
11	If-Desc	7.3.4
12	IPv6-Address List	7.3.5
13	Vendor	7.3.6
12-64534	Unassigned	
65535	Reserved	

Table 8: Sub-TLV Types

8.5. Error Codes

Value	Name	Remarks
0	Success	Success
1	Failure	Malformed message received.
2	TLV-Unknown	One or more of the TLVs was not understood.
3	TLV-Length	The TLV length is abnormal.
4-999	Unassigned	Unassigned basic error codes.

1000	Reserved	
1001	Version-Mismatch	The version negotiation fails. Terminate. The subsequent service processes corresponding to the UP are suspended.
1002	Keepalive Error	The keepalive negotiation fails.
1003	Timer Expires	The establishment timer expired.
1004-1999	Unassigned	Unassigned error codes for version negotiation.
2000	Reserved	
2001	Synch-NoReady	The data to be smoothed is not ready.
2002	Synch-Unsupport	The request for smooth data is not supported.
2003-2999	Unassigned	Unassigned data synchronization error codes.
3000	Reserved	
3001	Pool-Mismatch	The corresponding address pool cannot be found.
3002	Pool-Full	The address pool is fully allocated, and no address segment is available.
3003	Subnet-Mismatch	The address pool subnet cannot be found.
3004	Subnet-Conflict	Subnets in the address pool have been classified into other clients.
3005-3999	Unassigned	Unassigned error

		codes for address allocation.
4000	Reserved	
4001	Update-Fail-No-Res	The forwarding table fails to be delivered because the forwarding resources are insufficient.
4002	QoS-Update-Success	The QoS policy takes effect.
4003	QoS-Update-Sq-Fail	Failed to process the queue in the QoS policy.
4004	QoS-Update-CAR-Fail	Processing of the CAR in the QoS policy fails.
4005	Statistic-Fail-No-Res	Statistics processing failed due to insufficient statistics resources.
4006-4999	Unassigned	Unassigned forwarding table delivery error codes.
5000-4294967295	Reserved	

Table 9: Error Codes

8.6. If-Type Values

Defined values of the If-Type field in the If-Desc sub-TLV (see Section 7.3.4) are as follows:

Value	Meaning
0	Reserved
1	Fast Ethernet (FE)
2	GE
3	10GE
4	100GE

5	Eth-Trunk
6	Tunnel
7	VE
8-254	Unassigned
255	Reserved

Table 10: If-Type Values

8.7. Access-Mode Values

Defined values of the Access-Mode field in the BAS Function TLV (see Section 7.7) are as follows:

Value	Meaning
0	Layer 2 subscriber
1	Layer 3 subscriber
2	Layer 2 leased line
3	Layer 3 leased line
4-254	Unassigned
255	Reserved

Table 11: Access-Mode Values

8.8. Access Method Bits

Defined values of the Auth-Method4 and Auth-Method6 fields in the BAS Function TLV (see Section 7.7) are defined as bit fields as follows:

Bit	Meaning
0x01	PPPoE authentication
0x02	DOT1X authentication
0x04	Web authentication
0x08	Web fast authentication
0x10	Binding authentication
0x20	Reserved

0x40	Reserved
0x80	Reserved

Table 12: Auth-Method4 Values

Bit	Meaning
0x01	PPPoE authentication
0x02	DOT1X authentication
0x04	Web authentication
0x08	Web fast authentication
0x10	Binding authentication
0x20	Reserved
0x40	Reserved
0x80	Reserved

Table 13: Auth-Method6 Values

8.9. Route-Type Values

Values of the Route-Type field in the IPv4 and IPv6 Routing TLVs (see Sections 7.8.1 and 7.8.2) defined in this document are as follows:

Value	Meaning
0	User host route
1	Radius authorization FrameRoute
2	Network segment route
3	Gateway route
4	Radius authorized IP route
5	L2TP LNS side user route
6-65534	Unassigned
65535	Reserved

Table 14: Route-Type Values

8.10. Access-Type Values

Values of the Access-Type field in the Basic Subscriber TLV (see Section 7.9.1) defined in this document are as follows:

Value	Meaning
0	Reserved
1	PPP access (PPP [RFC1661])
2	PPP over Ethernet over ATM access (PPPoEoA)
3	PPP over ATM access (PPPoA [RFC3336])
4	PPP over Ethernet access (PPPoE [RFC2516])
5	PPPoE over VLAN access (PPPoEoVLAN [RFC2516])
6	PPP over LNS access (PPPoLNS)
7	IP over Ethernet DHCP access (IPoE_DHCP)
8	IP over Ethernet EAP authentication access (IPoE_EAP)
9	IP over Ethernet Layer 3 access (IPoE_L3)
10	IP over Ethernet Layer 2 Static access (IPoE_L2_STATIC)
11	Layer 2 Leased Line access (L2_Leased_Line)
12	Layer 2 VPN Leased Line access (L2VPN_Leased_Line)
13	Layer 3 Leased Line access (L3_Leased_Line)
14	Layer 2 Leased line Sub-User access (L2_Leased_Line_SUB_USER)
15	L2TP LAC tunnel access (L2TP_LAC)
16	L2TP LNS tunnel access (L2TP_LNS)
17-254	Unassigned
255	Reserved

Table 15: Access-Type Values

9. IANA Considerations

This document has no IANA actions.

10. Security Considerations

The Service, Control, and Management Interfaces between the CP and UP might be across the general Internet or other hostile environment. The ability of an adversary to block or corrupt messages or introduce spurious messages on any one or more of these interfaces would give the adversary the ability to stop subscribers from accessing network services, disrupt existing subscriber sessions, divert traffic, mess up accounting statistics, and generally cause havoc. Damage would not necessarily be limited to one or a few subscribers but could disrupt routing or deny service to one or more instances of the CP or otherwise cause extensive interference. If the adversary knows the details of the UP equipment and its forwarding rule capabilities, the adversary may be able to cause a copy of most or all user data to be sent to an address of the adversary's choosing, thus enabling eavesdropping.

Thus, appropriate protections MUST be implemented to provide integrity, authenticity, and secrecy of traffic over those interfaces. Whether such protection is used is the decision of the network operator. See [RFC6241] for Mi/NETCONF security. Security on the Si is dependent on the tunneling protocol used, which is out of scope for this document. Security for the Ci, over which S-CUSP flows, is further discussed below.

S-CUSP messages do not provide security. Thus, if these messages are exchanged in an environment where security is a concern, that security MUST be provided by another protocol such as TLS 1.3 [RFC8446] or IPsec. TLS 1.3 is the mandatory-to-implement protocol for interoperability. The use of a particular security protocol on the Ci is determined by configuration. Such security protocols need not always be used, and lesser security precautions might be appropriate because, in some cases, the communication between the CP and UP is in a benign environment.

11. References

11.1. Normative References

[RFC20] Cerf, V., "ASCII format for network interchange", STD 80, RFC 20, DOI 10.17487/RFC0020, October 1969, <https://www.rfc-editor.org/info/rfc20>.

[RFC793] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, DOI 10.17487/RFC0793, September 1981, <https://www.rfc-editor.org/info/rfc793>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC2661] Townsley, W., Valencia, A., Rubens, A., Pall, G., Zorn, G., and B. Palter, "Layer Two Tunneling Protocol "L2TP"", RFC 2661, DOI 10.17487/RFC2661, August 1999, <https://www.rfc-editor.org/info/rfc2661>.

[RFC2865] Rigney, C., Willens, S., Rubens, A., and W. Simpson, "Remote Authentication Dial In User Service (RADIUS)", RFC 2865, DOI 10.17487/RFC2865, June 2000, <https://www.rfc-editor.org/info/rfc2865>.

[RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <https://www.rfc-editor.org/info/rfc6241>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

11.2. Informative References

[802.1Q] IEEE, "IEEE Standard for Local and metropolitan area networks--Bridges and Bridged Networks", IEEE 802.1Q-2018, DOI 10.1109/IEEESTD.2018.8403927, July 2018, <https://doi.org/10.1109/IEEESTD.2018.8403927>.

[RFC1661] Simpson, W., Ed., "The Point-to-Point Protocol (PPP)", STD 51, RFC 1661, DOI 10.17487/RFC1661, July 1994, <https://www.rfc-editor.org/info/rfc1661>.

[RFC2131] Droms, R., "Dynamic Host Configuration Protocol", RFC 2131, DOI 10.17487/RFC2131, March 1997, <https://www.rfc-editor.org/info/rfc2131>.

[RFC2516] Mamakos, L., Lidl, K., Evarts, J., Carrel, D., Simone, D., and R. Wheeler, "A Method for Transmitting PPP Over Ethernet (PPPoE)", RFC 2516, DOI 10.17487/RFC2516, February 1999, <https://www.rfc-editor.org/info/rfc2516>.

[RFC2698] Heinanen, J. and R. Guerin, "A Two Rate Three Color Marker", RFC 2698, DOI 10.17487/RFC2698, September 1999, <https://www.rfc-editor.org/info/rfc2698>.

[RFC3022] Srisuresh, P. and K. Egevang, "Traditional IP Network Address Translator (Traditional NAT)", RFC 3022, DOI 10.17487/RFC3022, January 2001, <https://www.rfc-editor.org/info/rfc3022>.

[RFC3336] Thompson, B., Koren, T., and B. Buffam, "PPP Over Asynchronous Transfer Mode Adaptation Layer 2 (AAL2)", RFC 3336, DOI 10.17487/RFC3336, December 2002, <https://www.rfc-editor.org/info/rfc3336>.

[RFC5511] Farrel, A., "Routing Backus-Naur Form (RBNF): A Syntax Used to Form Encoding Rules in Various Routing Protocol Specifications", RFC 5511, DOI 10.17487/RFC5511, April 2009, <https://www.rfc-editor.org/info/rfc5511>.

[RFC7042] Eastlake 3rd, D. and J. Abley, "IANA Considerations and IETF Protocol and Documentation Usage for IEEE 802 Parameters", BCP 141, RFC 7042, DOI 10.17487/RFC7042,

October 2013, <https://www.rfc-editor.org/info/rfc7042>.

[RFC7348] Mahalingam, M., Dutt, D., Duda, K., Agarwal, P., Kreeger, L., Sridhar, T., Bursell, M., and C. Wright, "Virtual eXtensible Local Area Network (VXLAN): A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks", RFC 7348, DOI 10.17487/RFC7348, August 2014, <https://www.rfc-editor.org/info/rfc7348>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/info/rfc8446>.

[TR-384] Broadband Forum, "Cloud Central Office Reference Architectural Framework", BBF TR-384, January 2018.

[WT-459] Broadband Forum, "Control and User Plane Separation for a Disaggregated BNG", BBF WT-459, 2019.

Acknowledgements

The helpful comments and suggestions from the following individuals are hereby acknowledged:

* Loa Andersson

* Greg Mirsky

Contributors

Zhenqiang Li
China Mobile
32 Xuanwumen West Ave
Xicheng District
Beijing
100053
China

Email: lizhenqiang@chinamobile.com

Mach(Guoyi) Chen
Huawei Technologies
Huawei Bldg., No. 156 Beiqing Road
Beijing
100095
China

Email: mach.chen@huawei.com

Zhouyi Yu
Huawei Technologies

Email: yuzhouyi@huawei.com

Chengguang Niu
Huawei Technologies

Email: chengguang.niu@huawei.com

Zitao Wang
Huawei Technologies

Email: wangzitao@huawei.com

Jun Song
Huawei Technologies

Email: song.jun@huawei.com

Dan Meng
H3C Technologies
No. 1 Lixing Center
No. 8 Guangxun South Street
Chaoyang District
Beijing
100102
China

Email: mengdan@h3c.com

Hanlei Liu
H3C Technologies
No. 1 Lixing Center
No. 8 Guangxun South Street
Chaoyang District
Beijing
100102
China

Email: hanlei_liu@h3c.com

Victor Lopez
Telefonica
Spain

Email: victor.lopezalvarez@telefonica.com

Authors' Addresses

Shujun Hu
China Mobile
32 Xuanwumen West Ave
Xicheng District

Beijing
100053
China

Email: hushujun@chinamobile.com

Donald Eastlake 3rd
Futurewei Technologies
2386 Panoramic Circle
Apopka, FL 32703
United States of America

Phone: +1-508-333-2270
Email: d3e3e3@gmail.com

Fengwei Qin
China Mobile
32 Xuanwumen West Ave
Xicheng District
Beijing
100053
China

Email: qinfengwei@chinamobile.com

Tee Mong Chua
Singapore Telecommunications Limited
31 Exeter Road, #05-04 Comcentre Podium Block
SINGAPORE 239732
Singapore

Email: teemong@singtel.com

Daniel Huang
ZTE

Email: huang.guangping@zte.com.cn