

INTERNET-DRAFT  
Intended status: Proposed Standard

S. Hu  
China Mobile  
D. Eastlake  
Futurewei Technologies  
M. Chen  
Huawei Technologies  
F. Qin  
Z. Li  
China Mobile  
T. Chua  
Singapore Telecommunications  
D. Huang  
ZTE  
July 3, 2019

Expires: January 2, 2020

Control-Plane and User-Plane Separation BNG  
Simple Control Channel Protocol (S-CUSP)  
draft-cuspd-rtgwg-cu-separation-bng-protocol-06

## Abstract

This document specifies the Simple Control Plane (CP) and User Plane (UP) Separation Broadband Network Gateway (BNG) control channel Protocol (S-CUSP) for communications between a CP and a UP. S-CUSP is designed to be flexible and extensible so as to easily allow for the addition of further messages and data items to meet future requirements.

## Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Distribution of this document is unlimited. Comments should be sent to the authors or the RGTWG working group mailing list: [rtgwg@ietf.org](mailto:rtgwg@ietf.org).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

---

INTERNET-DRAFT

Simple BNG CUSP

The list of current Internet-Drafts can be accessed at  
<http://www.ietf.org/lid-abstracts.html>. The list of Internet-Draft  
Shadow Directories can be accessed at  
<http://www.ietf.org/shadow.html>.

INTERNET-DRAFT

Simple BNG CUSP

## Table of Contents

<a href="#">1. Introduction.....</a>	<a href="#">6</a>
<a href="#">2. Terminology.....</a>	<a href="#">7</a>
<a href="#">2.1 Implementation Requirement Keywords.....</a>	<a href="#">7</a>
<a href="#">2.2 Terms.....</a>	<a href="#">7</a>
<a href="#">3. BNG CUPS Overview.....</a>	<a href="#">10</a>
<a href="#">3.1 BNG CUPS Motivation.....</a>	<a href="#">10</a>
<a href="#">3.2 BNG CUPS Architecture Overview.....</a>	<a href="#">10</a>
<a href="#">3.3 BNG CUPS Interfaces.....</a>	<a href="#">12</a>
<a href="#">3.3.1 Service Interface.....</a>	<a href="#">13</a>
<a href="#">3.3.2 Control Interface.....</a>	<a href="#">14</a>
<a href="#">3.3.3 Management Interface.....</a>	<a href="#">14</a>
<a href="#">3.4 BNG CUPS Procedure Overview.....</a>	<a href="#">14</a>
<a href="#">4. S-CUSP Protocol Overview.....</a>	<a href="#">18</a>
<a href="#">4.1 Control Channel Related Procedures.....</a>	<a href="#">18</a>
<a href="#">4.1.1 S-CUSP Session Establishment.....</a>	<a href="#">18</a>
<a href="#">4.1.2 Keep Alive.....</a>	<a href="#">19</a>
<a href="#">4.2 Node Related Procedures.....</a>	<a href="#">20</a>
<a href="#">4.2.1 UP Resource Report.....</a>	<a href="#">20</a>
<a href="#">4.2.2 Update BAS Function on Access Interface.....</a>	<a href="#">21</a>
<a href="#">4.2.3 Update Network Routing.....</a>	<a href="#">21</a>
<a href="#">4.2.4 CGN Public IP Address Allocation.....</a>	<a href="#">22</a>
<a href="#">4.2.5 Data Synchronization between the CP and UP.....</a>	<a href="#">23</a>
<a href="#">4.3 Subscriber Session Related Procedures.....</a>	<a href="#">24</a>
<a href="#">4.3.1 Create Subscriber Session.....</a>	<a href="#">25</a>
<a href="#">4.3.2 Update Subscriber Session.....</a>	<a href="#">26</a>
<a href="#">4.3.3 Delete Subscriber Session.....</a>	<a href="#">27</a>

4.3.4	Subscriber Session Events Report.....	27
5.	S-CUSP Call Flows.....	29
5.1	IPoE.....	29
5.1.1	DHCPv4 Access.....	29
5.1.2	DHCPv6 Access.....	30
5.1.3	IPv6 SLAAC Access.....	32
5.1.4	DHCPv6 + SLAAC Access.....	33
5.1.5	DHCP Dual Stack Access.....	35
5.1.6	L2 Static Subscriber Access.....	37
5.2	PPPoE.....	40
5.2.1	IPv4 PPPoE Access.....	40
5.2.2	IPv6 PPPoE Access.....	41
5.2.3	PPPoE Dual Stack Access.....	43
5.3	WLAN Access.....	45
5.4	L2TP.....	47
5.4.1	L2TP LAC Access.....	47
5.4.2	L2TP LNS IPv4 Access.....	49
5.4.3	L2TP LNS IPv6 Access.....	51
5.5	CGN (Carrier Grade NAT).....	54

## Table of Contents (continued)

5.6	L3 Leased Line Access.....	55
5.6.1	Web Authentication.....	55
5.6.2	User Traffic Trigger.....	57
5.7	Multicast Access.....	58
6.	S-CUSP Message Formats.....	60
6.1	Common Message Header.....	60
6.2	Control Messages.....	61
6.2.1	Hello Message.....	61
6.2.2	Keepalive Message.....	62
6.2.3	Sync_Request Message.....	62
6.2.4	Sync_Begin Message.....	62
6.2.5	Sync_Data Message.....	63
6.2.6	Sync_End Message.....	63
6.2.7	Update_Request Message.....	64
6.2.8	Update_Response Message.....	64
6.3	Event Message.....	65
6.4	Report Message.....	66
6.5	CGN Messages.....	66
6.5.1	Addr_Allocation_Req Message.....	66
6.5.2	Addr_Allocation_Ack Message.....	66

<a href="#">6.5.3</a>	<a href="#">Addr_Renew_Req Message.....</a>	<a href="#">67</a>
<a href="#">6.5.4</a>	<a href="#">Addr_Renew_Ack Message.....</a>	<a href="#">67</a>
<a href="#">6.5.5</a>	<a href="#">Addr_Release_Req Message.....</a>	<a href="#">67</a>
<a href="#">6.5.6</a>	<a href="#">Addr_Release_Ack Message.....</a>	<a href="#">67</a>
<a href="#">6.6</a>	<a href="#">Vendor Message.....</a>	<a href="#">67</a>
<a href="#">6.7</a>	<a href="#">Error Message.....</a>	<a href="#">68</a>
<a href="#">7.</a>	<a href="#">S-CUSP TLVs and Sub-TLVs.....</a>	<a href="#">69</a>
<a href="#">7.1</a>	<a href="#">Common TLV Header.....</a>	<a href="#">69</a>
<a href="#">7.2</a>	<a href="#">Basic Data Fields.....</a>	<a href="#">70</a>
<a href="#">7.3</a>	<a href="#">Sub-TLV Format and Sub-TLVs.....</a>	<a href="#">71</a>
<a href="#">7.3.1</a>	<a href="#">Name sub-TLVs.....</a>	<a href="#">71</a>
<a href="#">7.3.2</a>	<a href="#">Ingress-CAR sub-TLV.....</a>	<a href="#">72</a>
<a href="#">7.3.3</a>	<a href="#">Egress-CAR sub-TLV.....</a>	<a href="#">72</a>
<a href="#">7.3.4</a>	<a href="#">If-Desc sub-TLV.....</a>	<a href="#">73</a>
<a href="#">7.3.5</a>	<a href="#">IPv6 Address List sub-TLV.....</a>	<a href="#">75</a>
<a href="#">7.3.6</a>	<a href="#">Vendor sub-TLV.....</a>	<a href="#">75</a>
<a href="#">7.4</a>	<a href="#">The Hello TLV.....</a>	<a href="#">77</a>
<a href="#">7.5</a>	<a href="#">The Keep Alive TLV.....</a>	<a href="#">78</a>
<a href="#">7.6</a>	<a href="#">The Error Information TLV.....</a>	<a href="#">79</a>
<a href="#">7.7</a>	<a href="#">BAS Function TLV.....</a>	<a href="#">79</a>
<a href="#">7.8</a>	<a href="#">Routing TLVs.....</a>	<a href="#">82</a>
<a href="#">7.8.1</a>	<a href="#">IPv4 Routing TLV.....</a>	<a href="#">82</a>
<a href="#">7.8.2</a>	<a href="#">IPv6 Routing TLV.....</a>	<a href="#">84</a>
<a href="#">7.9</a>	<a href="#">Subscriber TLVs.....</a>	<a href="#">85</a>
<a href="#">7.9.1</a>	<a href="#">Basic Subscriber TLV.....</a>	<a href="#">86</a>
<a href="#">7.9.2</a>	<a href="#">PPP Subscriber TLV.....</a>	<a href="#">88</a>
<a href="#">7.9.3</a>	<a href="#">IPv4 Subscriber TLV.....</a>	<a href="#">89</a>

## Table of Contents (continued)

<a href="#">7.9.4</a>	<a href="#">IPv6 Subscriber TLV.....</a>	<a href="#">90</a>
<a href="#">7.9.5</a>	<a href="#">IPv4 Static Subscriber Detect TLV.....</a>	<a href="#">91</a>
<a href="#">7.9.6</a>	<a href="#">IPv6 Static Subscriber Detect TLV.....</a>	<a href="#">93</a>
<a href="#">7.9.7</a>	<a href="#">L2TP-LAC Subscriber TLV.....</a>	<a href="#">94</a>
<a href="#">7.9.8</a>	<a href="#">L2TP-LNS Subscriber TLV.....</a>	<a href="#">95</a>
<a href="#">7.9.9</a>	<a href="#">L2TP-LAC Tunnel TLV.....</a>	<a href="#">95</a>
<a href="#">7.9.10</a>	<a href="#">L2TP-LNS Tunnel TLV.....</a>	<a href="#">96</a>
<a href="#">7.9.11</a>	<a href="#">Update Response TLV.....</a>	<a href="#">97</a>
<a href="#">7.9.12</a>	<a href="#">Subscriber Policy TLV.....</a>	<a href="#">98</a>
<a href="#">7.9.13</a>	<a href="#">Subscriber CGN Port Range TLV.....</a>	<a href="#">100</a>
<a href="#">7.10</a>	<a href="#">Device Status TLVs.....</a>	<a href="#">100</a>
<a href="#">7.10.1</a>	<a href="#">Interface Status TLV.....</a>	<a href="#">101</a>
<a href="#">7.10.2</a>	<a href="#">Board Status TLV.....</a>	<a href="#">101</a>

<a href="#">7.11</a>	<a href="#">CGN TLVs.....</a>	<a href="#">102</a>
<a href="#">7.11.1</a>	<a href="#">Address Allocation Request TLV.....</a>	<a href="#">102</a>
<a href="#">7.11.2</a>	<a href="#">Address Allocation Response TLV.....</a>	<a href="#">103</a>
<a href="#">7.11.3</a>	<a href="#">Address Renewal Request TLV.....</a>	<a href="#">104</a>
<a href="#">7.11.4</a>	<a href="#">The Address Renewal Response TLV.....</a>	<a href="#">105</a>
<a href="#">7.11.5</a>	<a href="#">Address Release Request TLV.....</a>	<a href="#">106</a>
<a href="#">7.11.6</a>	<a href="#">The Address Release Response TLV.....</a>	<a href="#">106</a>
<a href="#">7.12</a>	<a href="#">Event TLVs.....</a>	<a href="#">107</a>
<a href="#">7.12.1</a>	<a href="#">Subscriber Traffic Statistics TLV.....</a>	<a href="#">108</a>
<a href="#">7.12.2</a>	<a href="#">Subscriber Detection Result TLV.....</a>	<a href="#">109</a>
<a href="#">7.13</a>	<a href="#">Vendor TLV.....</a>	<a href="#">110</a>
<a href="#">8</a>	<a href="#">Implementation Status.....</a>	<a href="#">112</a>
<a href="#">8.1</a>	<a href="#">Implementations.....</a>	<a href="#">112</a>
<a href="#">8.1.1</a>	<a href="#">Huawei Technologies.....</a>	<a href="#">112</a>
<a href="#">8.1.2</a>	<a href="#">ZTE.....</a>	<a href="#">113</a>
<a href="#">8.1.3</a>	<a href="#">H3C.....</a>	<a href="#">113</a>
<a href="#">8.2</a>	<a href="#">Hackathon.....</a>	<a href="#">113</a>
<a href="#">8.3</a>	<a href="#">EANTC Testing.....</a>	<a href="#">114</a>
<a href="#">9</a>	<a href="#">IANA Considerations.....</a>	<a href="#">115</a>
<a href="#">9.1</a>	<a href="#">Message Types.....</a>	<a href="#">115</a>
<a href="#">9.2</a>	<a href="#">TLV Types.....</a>	<a href="#">115</a>
<a href="#">9.3</a>	<a href="#">TLV Operation Codes.....</a>	<a href="#">117</a>
<a href="#">9.4</a>	<a href="#">Sub-TLV Types.....</a>	<a href="#">118</a>
<a href="#">9.5</a>	<a href="#">Error Codes.....</a>	<a href="#">118</a>
<a href="#">10</a>	<a href="#">Security Considerations.....</a>	<a href="#">120</a>
	<a href="#">Contributors.....</a>	<a href="#">121</a>
	<a href="#">Normative References.....</a>	<a href="#">122</a>
	<a href="#">Informative References.....</a>	<a href="#">123</a>
	<a href="#">Authors' Addresses.....</a>	<a href="#">125</a>

## [1](#). Introduction

A fixed network Broadband Network Gateway (BNG) is an Ethernet-centric IP edge router, and the aggregation point for user traffic. To provide centralized session management, flexible address allocation, high scalability for subscriber management capacity, and cost-efficient redundancy, the Control/User (CU) separated BNG

framework is described in [[TR-384](#)]. The CU separated service Control Plane (CP), which is responsible for user access authentication and setting forwarding entries in User Planes (UPs), can be virtualized and centralized. The routing control and forwarding plane, i.e. the BNG user plane (local), can be distributed across the infrastructure. Other structures can also be supported such as both CP and UP being virtual or both being physical.

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

## [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 (BRAS, BNG).

BNG: Broadband Network Gateway. A broadband remote access server (BRAS (BRoadband Access Server), B-RAS or BBRAS) 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. BRAS can also be referred to as a Broadband Network Gateway (BNG).

BRAS: BRoadband Access Server (BNG).

CAR: Committed Access Rate.

CBS: Committed Burst Size.

CGN: Carrier Grade NAT.

Ci: Control Interface.

CIR: Committed Information Rate.

CoA: Change of Authorization.



---

INTERNET-DRAFT

Simple BNG CUSP

CP: Control Plane.

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

CPE: Customer Premises Equipment.

CU: Control-plane / User-plane.

CUSP: Control and User plane Separation Protocol.

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

DHCP: Dynamic Host Configuration Protocol [[RFC2131](#)].

dial-up: This refers to the initial connection messages when a new user appears. The name is left over from when users 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.

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, 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 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 rectify faults rapidly.

**Slow service provisioning:** The coupling of control plane and forwarding plane, in addition to a distributed network control mechanism, means that any new technology has to rely heavily on

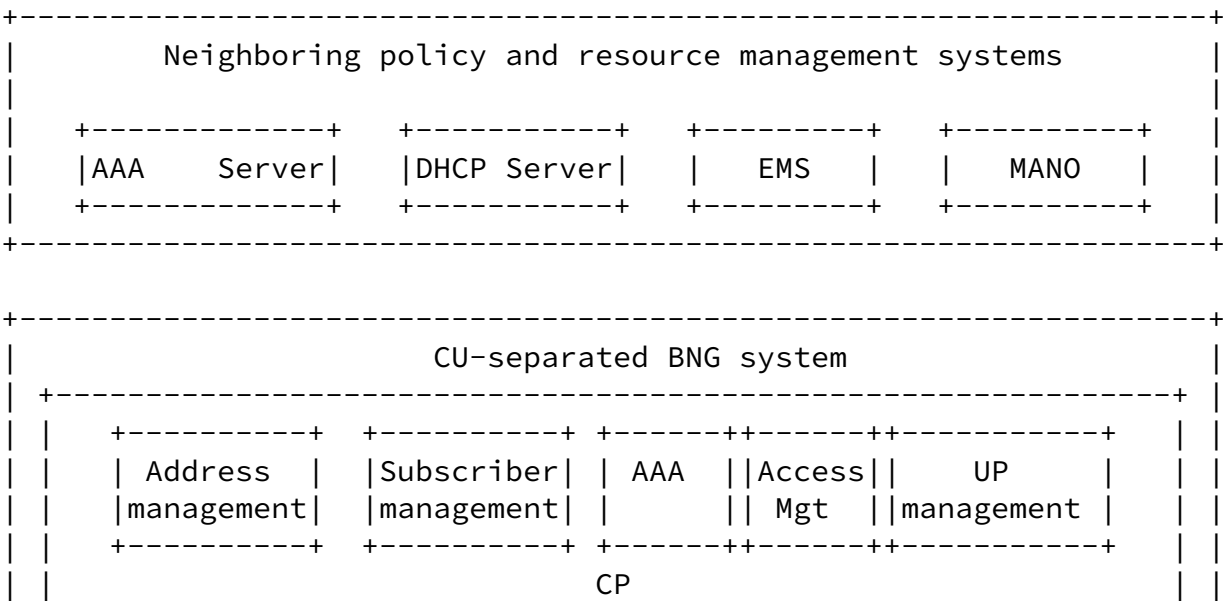
the existing network devices.

To address these challenges for fixed networks, the framework for a cloud-based BNG with Control Plane and User Plane (CU) separation 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 Control Plane (CP). And the traditional router's Control Plane and Forwarding Plane are both preserved on BNG devices in the form of a User Plane (UP).

3.2 BNG CUPS Architecture Overview

The functions in a traditional BNG can be divided into two parts: one is the user access management function, the other is the router function. The user management function can be centralized and deployed as a concentrated module or device, called the BNG Control Plane (BNG-CP). The other functions, such as the router function and forwarding engine, can be deployed in the form of the BNG User Plane (BNG-UP).

The following figure shows the architecture of CU separated BNG:



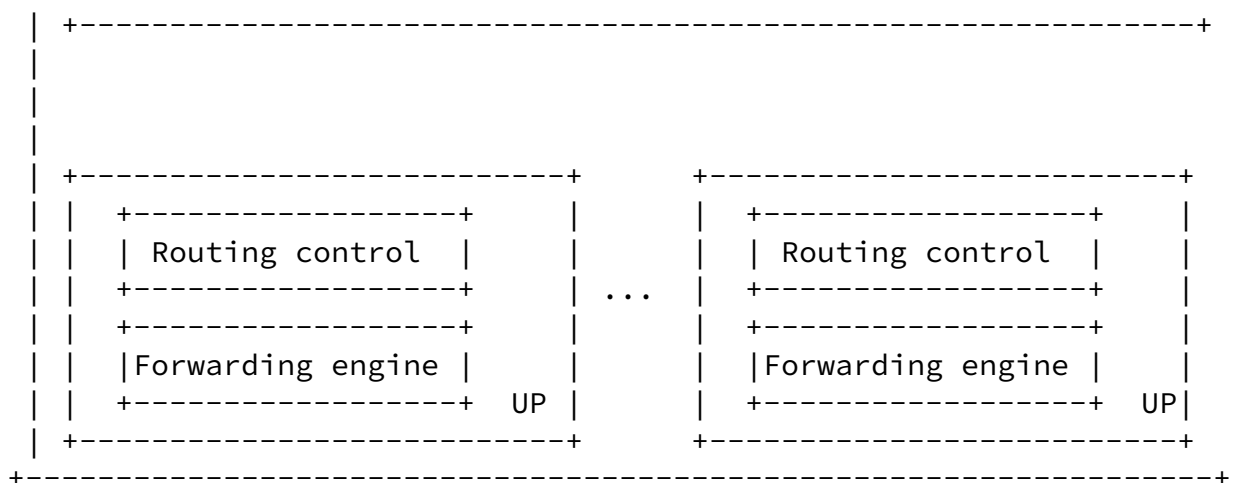


Figure 1: Architecture of CU Separated BNG

As shown in Figure 1, the BNG Control Plane could be virtualized and centralized, which provides benefits such as centralized session management, flexible address allocation, high scalability for subscriber management capacity, and cost-efficient redundancy, etc. The functional components inside the BNG Service Control Plane can be implemented as Virtual Network Functions (VNFs) and hosted in a Network Function Virtualization Infrastructure (NFVI).

The User Plane Management module in the BNG Control Plane centrally manages the distributed BNG User Planes (e.g. load balancing), as well as the setup, deletion, and maintenance of channels between Control Planes and User Planes. Other modules in the BNG control plane, such as address management, AAA, etc., are responsible for the connection with outside subsystems in order to fulfill those services. Note that the User Plane SHOULD support both physical and virtual network functions. For example, BNG user plane L3 forwarding

related network functions can be disaggregated and distributed across the physical infrastructure. And the other control plane and management plane functions in the CU Separation BNG can be moved into the NFVI for virtualization [\[TR-384\]](#).

The details of CU separated BNG's function components are as following:

The Control Plane is responsible for the following:

1. Address management: unified address pool management and CGN subscriber address traceability management.
2. 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 implement accounting, data capacity limitation, and QoS enforcement policies.
3. Subscriber management: user entry management and forwarding policy management.
4. Access management: process user dial-up packets, such as PPPoE, DHCP, L2TP, etc.
5. UP management: management of UP interface status, and the setup, deletion, and maintenance of channels between CP and UP.

The User Plane is responsible for the following:

1. Routing control functions: responsible for constructing routing forwarding plane (e.g., routing, multicast, MPLS, etc.).
2. Routing and Service Forwarding plane functions: responsible including traffic forwarding, QoS and traffic statistics collection.

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

### [3.3](#) BNG CUPS Interfaces

To support the communication between the Control Plane and User Plane, three interfaces are assumed. These are referred to as the Service Interface (Si), Control Interface (Ci), and Management Interface (Mi) as shown in Figure 2.

```

+-----+
|               |
+-----+

```

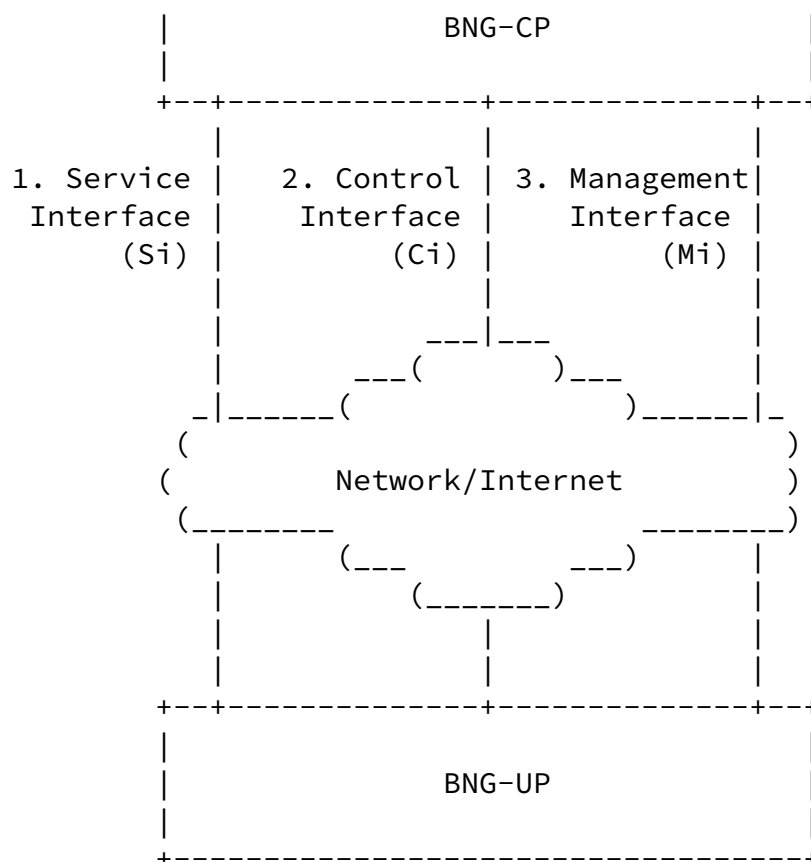


Figure 2: Interfaces Between the CP and UP of the BNG

### 3.3.1 Service Interface

For a traditional BNG (without CU separation), the user dial-up signals are terminated and processed by the control plane 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 Service Interface (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 VXLAN [\[RFC7348\]](#).

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

### 3.3.2 Control Interface

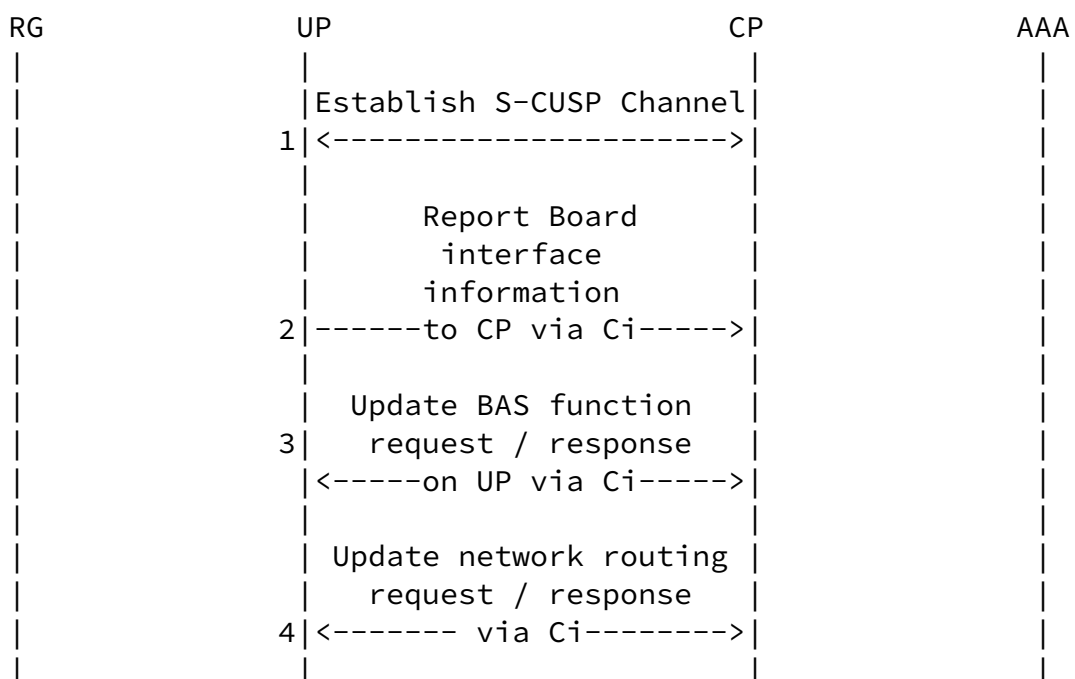
The CP uses the Control Interface to deliver subscriber session states, network routing entries, etc. to the UP (see [Section 6.2.7](#)). The UP uses this interface to report subscriber service statistics, subscriber detection results, etc. to the CP (see [Sections 6.3](#) and [6.4](#)). A carrying protocol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

### 3.3.3 Management Interface

NETCONF [[RFC6241](#)] is the protocol used on the Management Interface between a CP and UP. It is used to configure the parameters of the Control Interface, Service Interface, the 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 are out of scope for this document.

## 3.4 BNG CUPS Procedure Overview

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





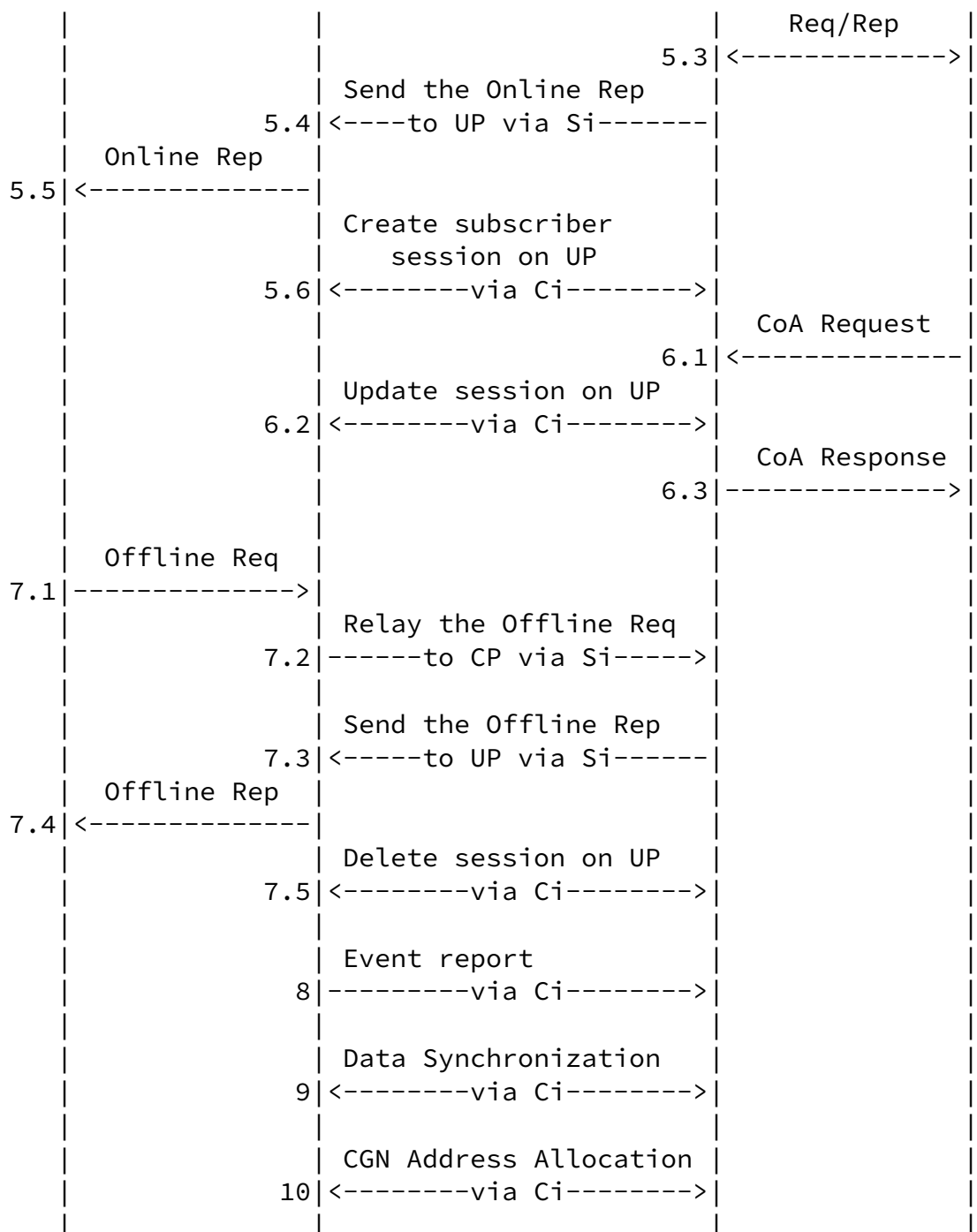
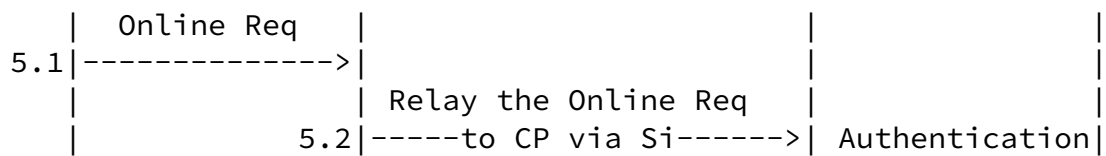


Figure 3: BNG CUPS Procedures Overview

1. S-CUSP session establishment: This is the first step of BNG CUPS procedures. Once the Control Interface parameters are configured on a UP. It will start to setup 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 side interfaces of this UP to the CP. A board can also be called a Line/Service Process

Unit (LPU/SPU) card. The subscriber side interfaces refer to the interfaces that connect the Access Network nodes (e.g., OLT: Optical Line Terminal, DSLAM: Digital Subscriber Line Access Multiplexer, etc.). The CP can use this information to enable the Broadband Access Service (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 (Broadband Access Service) 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 process. When a UP receives a dial-up request, it will relay the request packet to a CP through the Service Interface. 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, which enables 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 detail 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 Service Interface. The CP will send back a response to the UP through the Service Interface. The UP will then forward the offline response to the subscriber. Then the CP will delete the session on the UP through the Control Interface.

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 Sections [4.2.5](#) for more detail on CP and UP Synchronization.
10. CGN address allocation: See Sections [4.2.4](#) for more detail on CGN Address Allocation.

## [4. S-CUSP Protocol Overview](#)

### [4.1 Control Channel Related 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, one called the Master CP and the other called the 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.

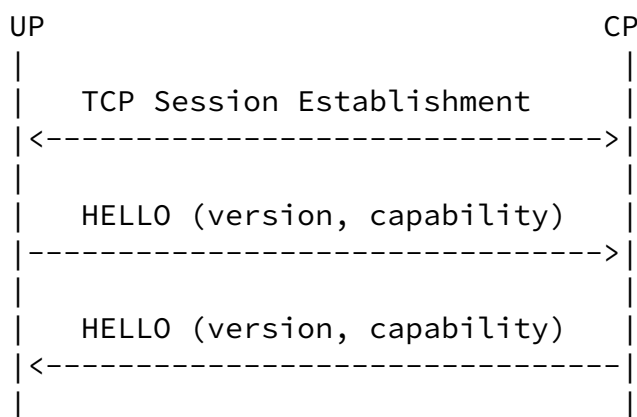


Figure 4: S-CUSP Session Establishment

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

1. Establishment of a TCP [[RFC793](#)] connection (3-way handshake) between the CP and the UP using a configured port from the dynamic port range (49152-65535).
2. Establishment of a 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. So, the version negotiation is based on whether a version can be support by both the CP and the UP. For a CP or UP, if a Hello message is received that

does not indicate a version supported by both, a subsequent Hello message with an Error Information TLV will be sent to the peer 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 Dead Timer field. The CP and UP have to agree on the Keepalive Timer and Dead Timer. 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 back-off session establishment retry procedure.

The S-CUSP session timer values that need to be configured are summarized in the table below.

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

#### [4.1.2](#) Keep Alive

Once an S-CUSP session has been established, a UP or CP may want to know that its S-CUSP peer is still available for use.

Each end of a 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.

The ends of the S-CUSP session also run DeadTimers, and they restart the timers whenever a message is received on the session. If one end of the session receives no message after the DeadTimer expires, it declares the session dead. The session will be closed.

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 timer may be disabled by setting it to zero.

The recommended default for the DeadTimer is 4 times the value of the Keepalive timer used by the remote peer. This implies there is essentially no risk of TCP congestion due to excessive Keepalive messages.

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

## [4.2](#) Node Related Procedures

### [4.2.1](#) UP Resource Report

Once an S-CUSP session has been established between a CP and an UP. The UP reports the information of the Boards and access side interfaces on this 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 Broadband Access Service (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 an UP may trigger the CP to start a warm-standby process, by moving the on-line subscriber sessions to a standby UP and then direct the affected subscribers to access the Internet through the standby UP.

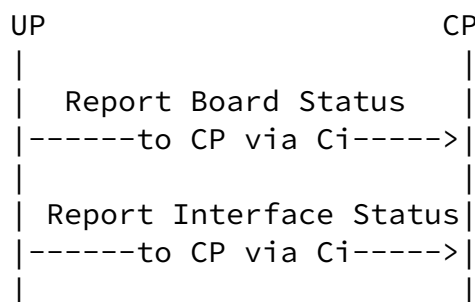


Figure 5: UP Board and Interface Report

Board status information is carried in the Board Status TLV ([Section 7.10.2](#)) and Interface status information is carried in Interface Status TLV ([Section 7.10.1](#)). Both Board 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/de-activate/modify the BAS functions on specified interfaces through the Update\_Request and Update\_Response message ([Section 6.2](#)) exchanges carrying the BAS Function TLV ([Section 7.7](#)).

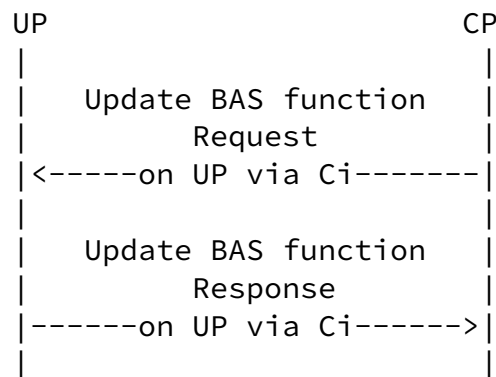
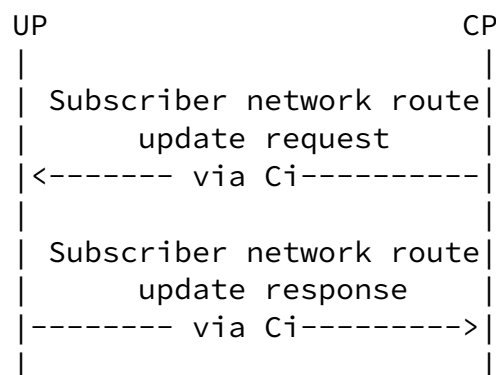


Figure 6: Update BAS Function

#### 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 allocated 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 subscriber network routing update request and response are achieved through the Update Request and Response Message exchanges by carrying the IPv4/IPv6 Routing Information TLVs ([Section 7.8](#)).

### [4.2.4](#) CGN Public IP Address Allocation

The following sequences describe the CGN address management related procedures. 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 a 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 in the IPv4/IPv6 Routing TLV (see [Section 7.1](#)) determines whether the request is an update or withdraw.

The relevant messages are defined in [Section 6.5](#).

INTERNET-DRAFT

Simple BNG CUSP

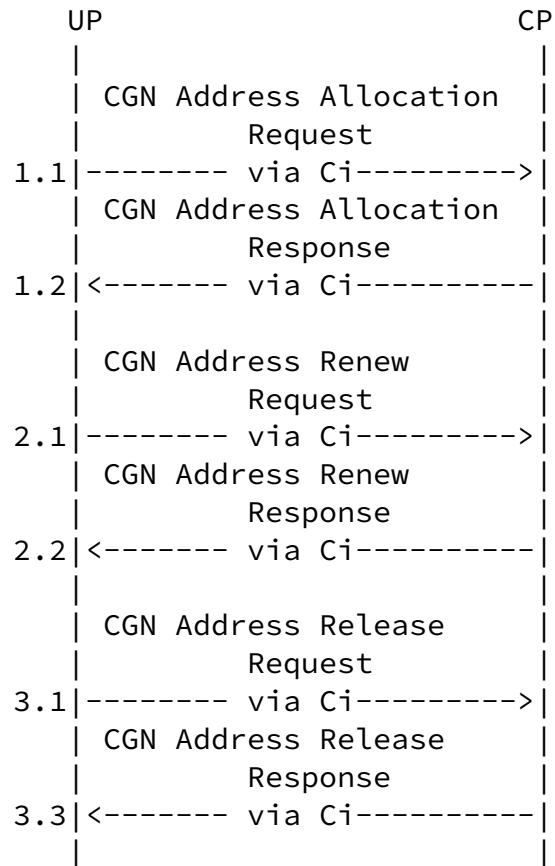


Figure 8: CGN Public IP Address Allocation

#### [4.2.5](#) Data Synchronization between the CP and UP

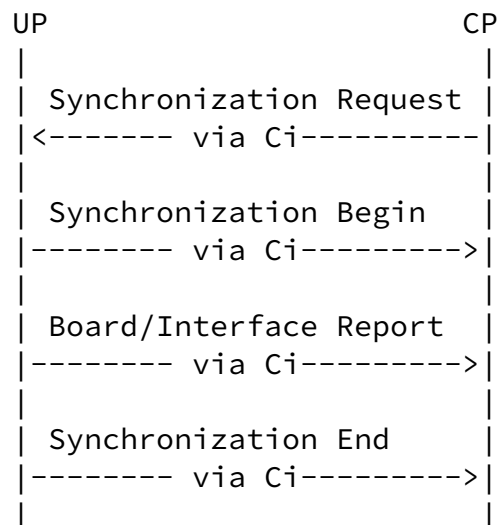
For a CU separated BNG, the UP will continue to function using the state that has been installed in it even if the CP fails or the session between the UP and CP fails.

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 finished, a Sync\_End message will be sent.

The following figure shows the process of data synchronization between a UP and a CP.



### 1) Synchronization from UP to CP

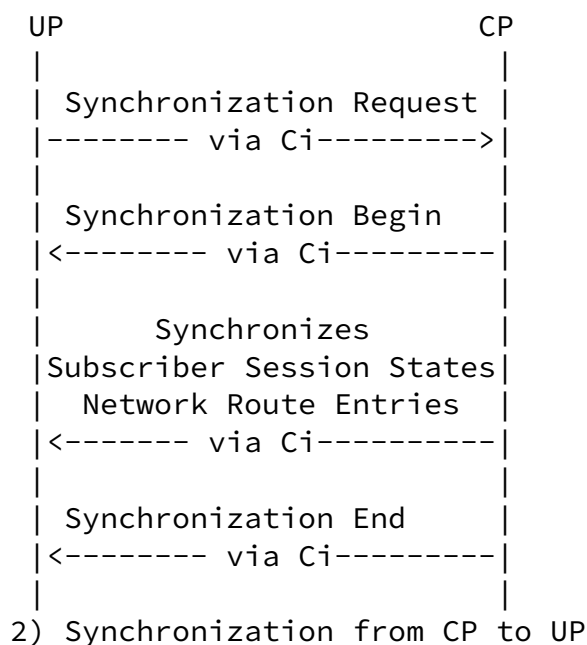


Figure 9: Data Synchronization

## 4.3 Subscriber Session Related 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 set of hardware resource have to be allocated (e.g., NP, TCAM etc.) to a session.

Subscriber session related procedures include subscriber session

create, update, delete, and statistics report. The following sub-sections give a high level view of the procedures.

### 4.3.1 Create Subscriber Session

The below sequence 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.](#))

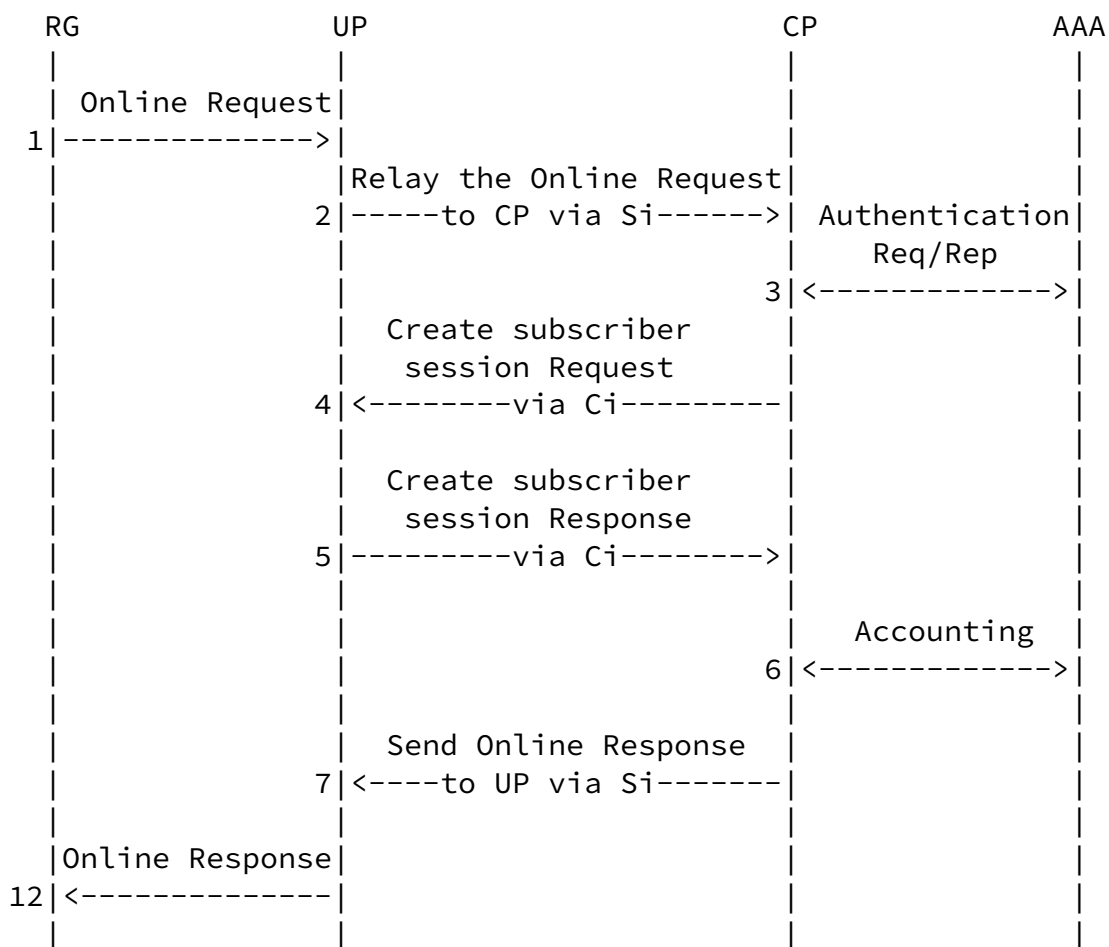


Figure 10: Subscriber Session Create

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 Service Interface (Step 2). The Service Interface is implemented by a tunneling technology.

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 sever, 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 session create request to the UP through the Control Interface (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). And the UP will forward the Online Response to the RG (step 8).

This completes the subscriber online process.

#### [4.3.2](#) Update Subscriber Session

The following numbered sequence shows the process of subscriber session update.

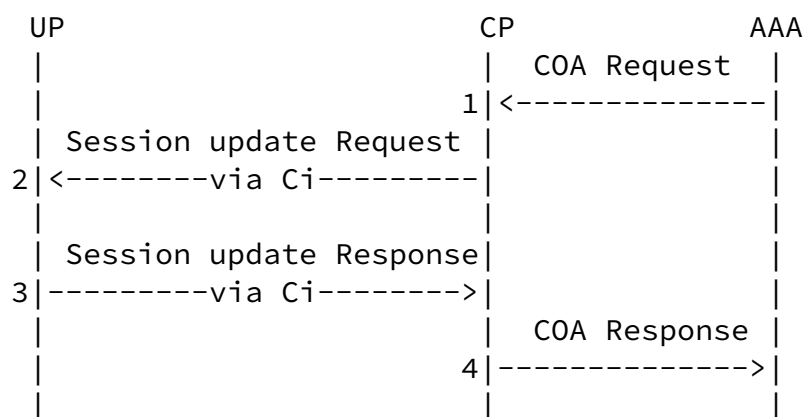


Figure 11: Subscriber Session Update

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 Control Interface.

#### 4.3.3 Delete Subscriber Session

The below call flow shows generally how S-CUPS deals with a subscriber offline request.

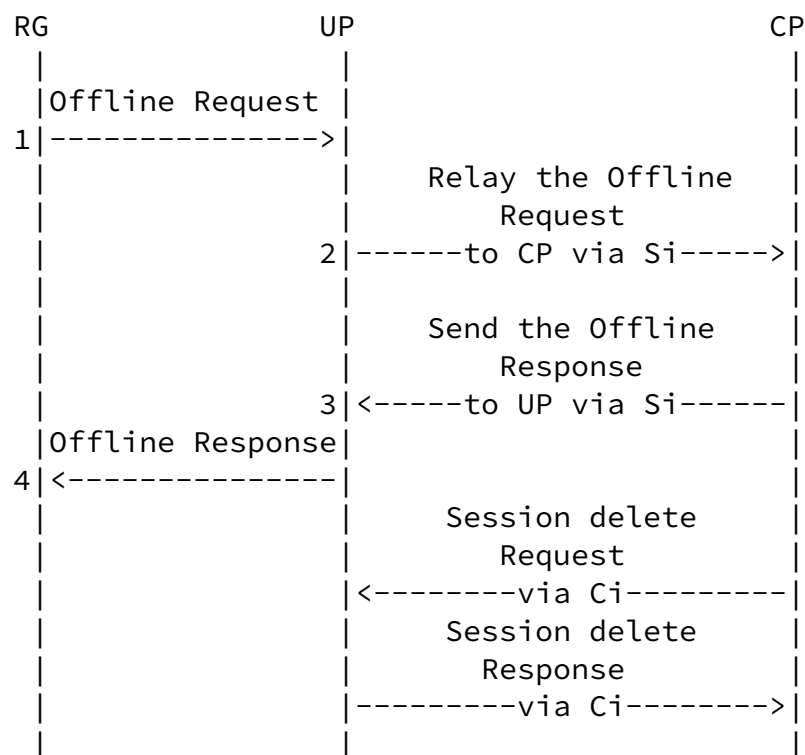


Figure 12: Subscriber Session Delete

Similar to the session creation process, when a UP receives an offline request from a 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. Then, it sends a reply to the UP through the Service Interface 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

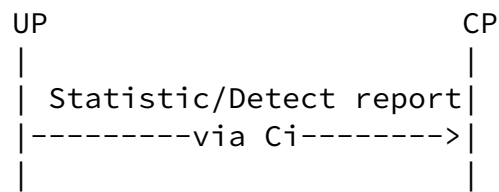


Figure 13: Events Report

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



5. S-CUSP Call Flows

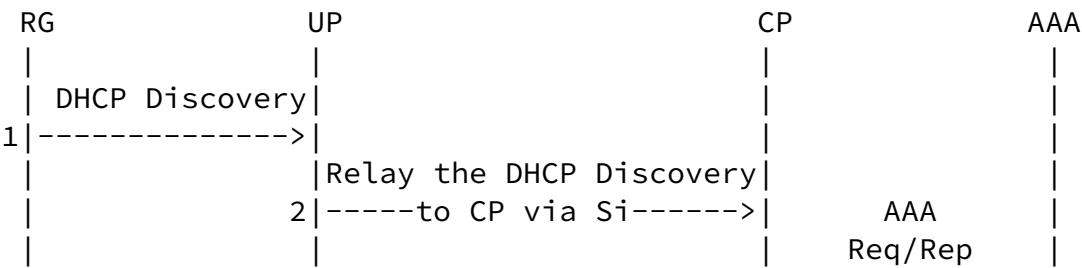
The subsections below give an overview of various "dial-up" interactions over the Service Interface followed by an overview of the setting of various information in the UP by the CP using S-CUSP over the Control Interface.

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 shows detailed procedures for DHCPv4 access.





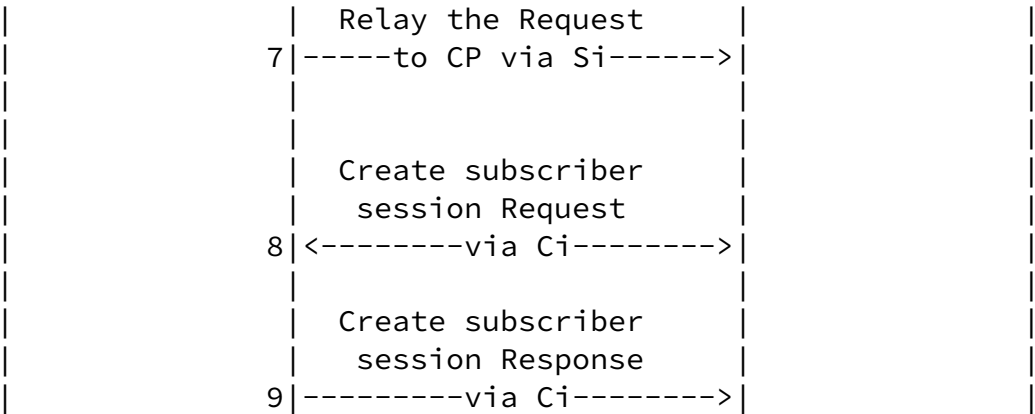
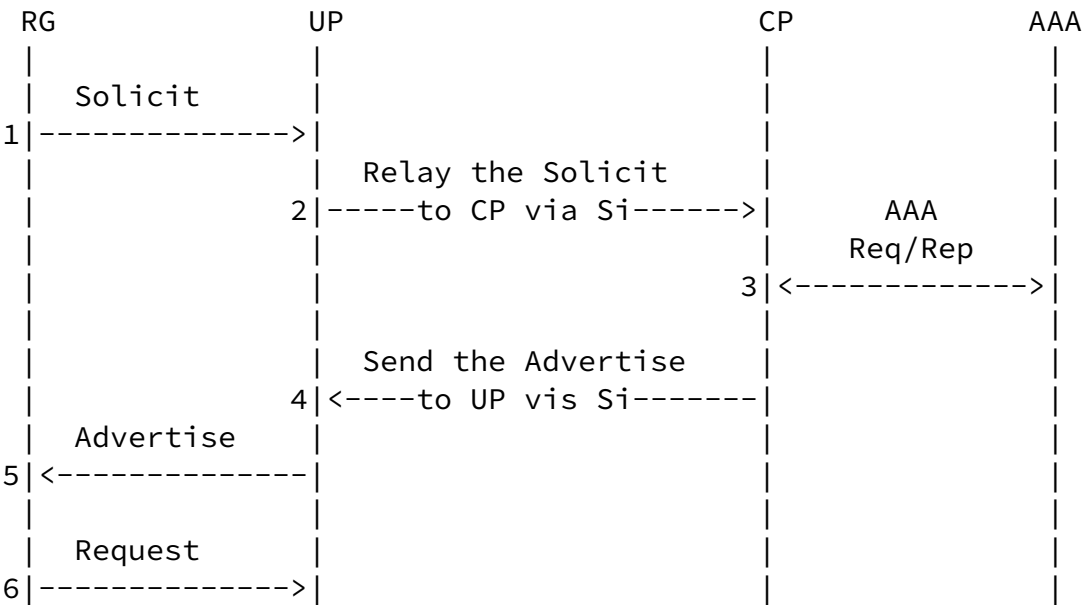
[<Subscriber Policy TLV>]

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 shows detailed procedures for DHCPv6 access.



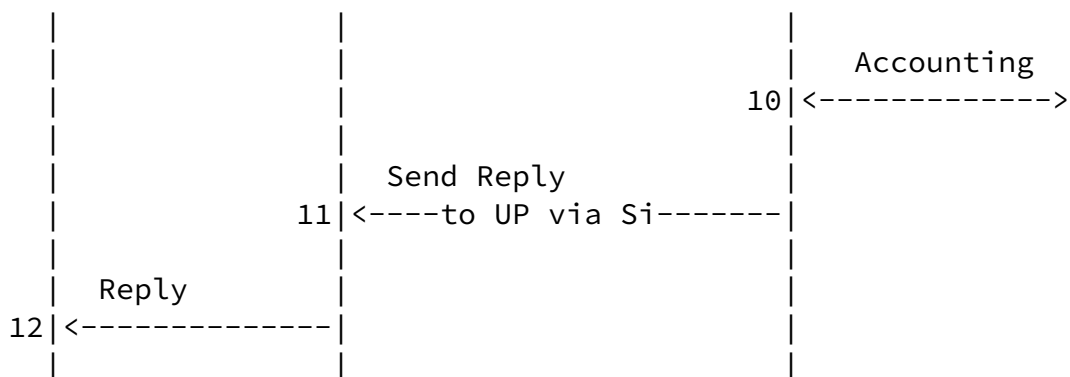


Figure 15: DHCPv6 Access

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 SLAAC Access

The following flow shows the IPv6 SLAAC access process.

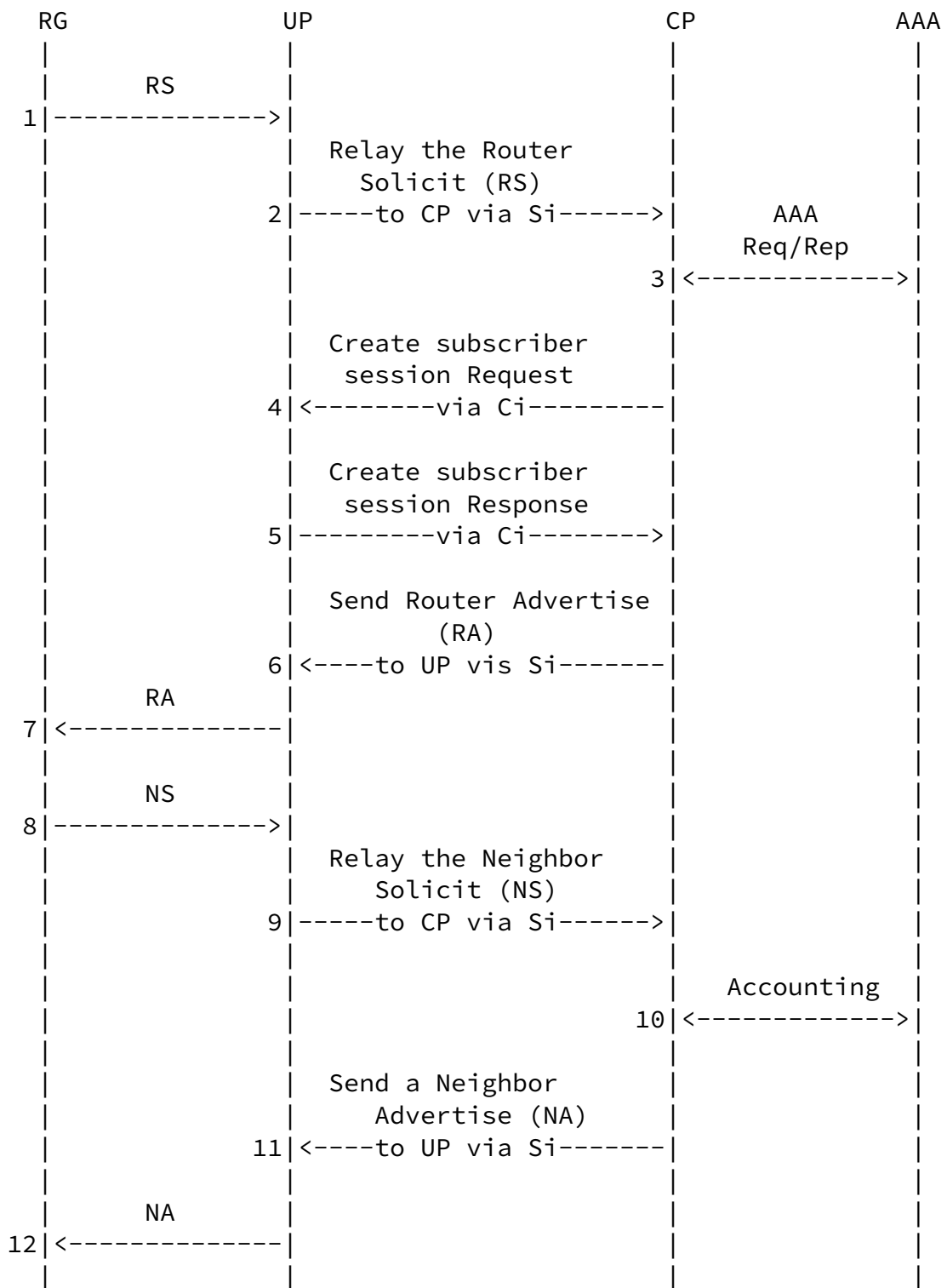


Figure 16: IPv6 SLAAC Access

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:

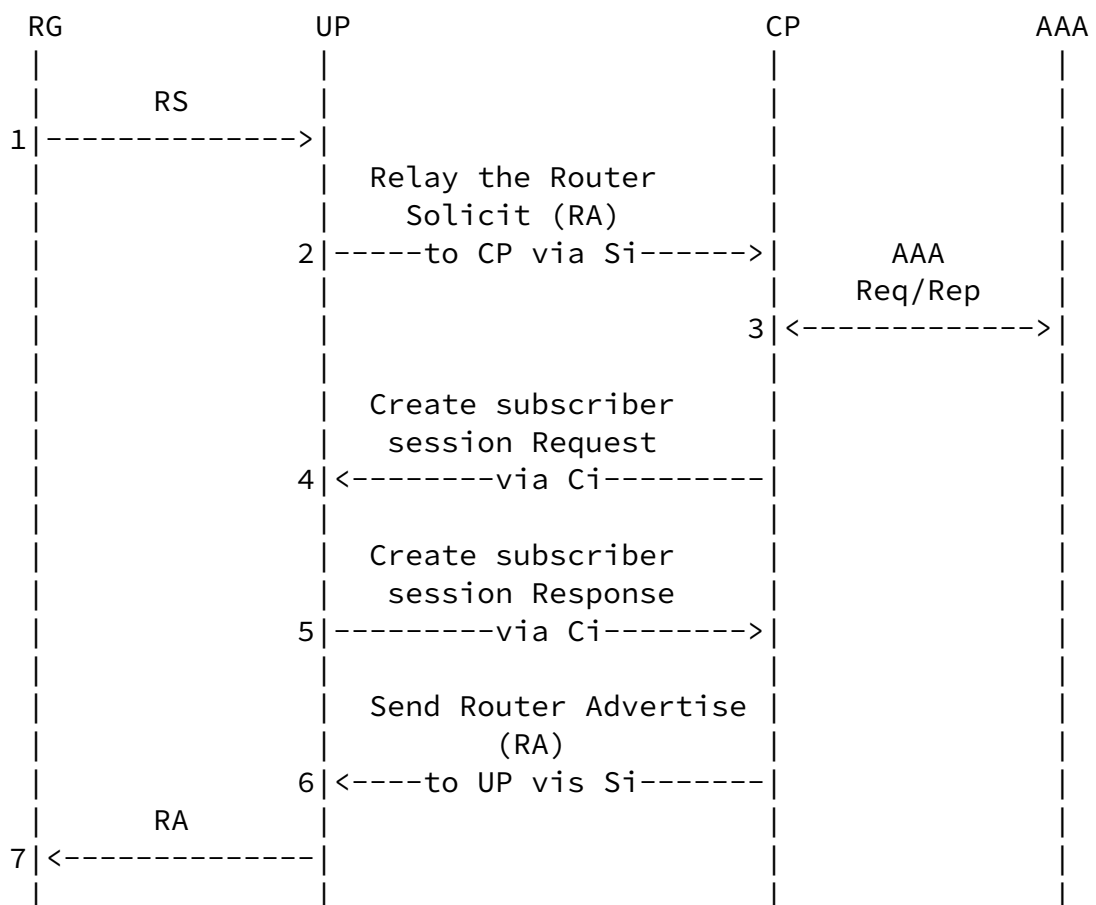
```
<Update_Request Message> ::= <Common Header>
                               <Basic Subscriber TLV>
                               <IPv6 Subscriber TLV>
                               <IPv6 Routing TLV>
                               [<Subscriber Policy TLV>]
```

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 + SLAAC Access

The following call flow shows the DHCP IPv6 and SLAAC access process.



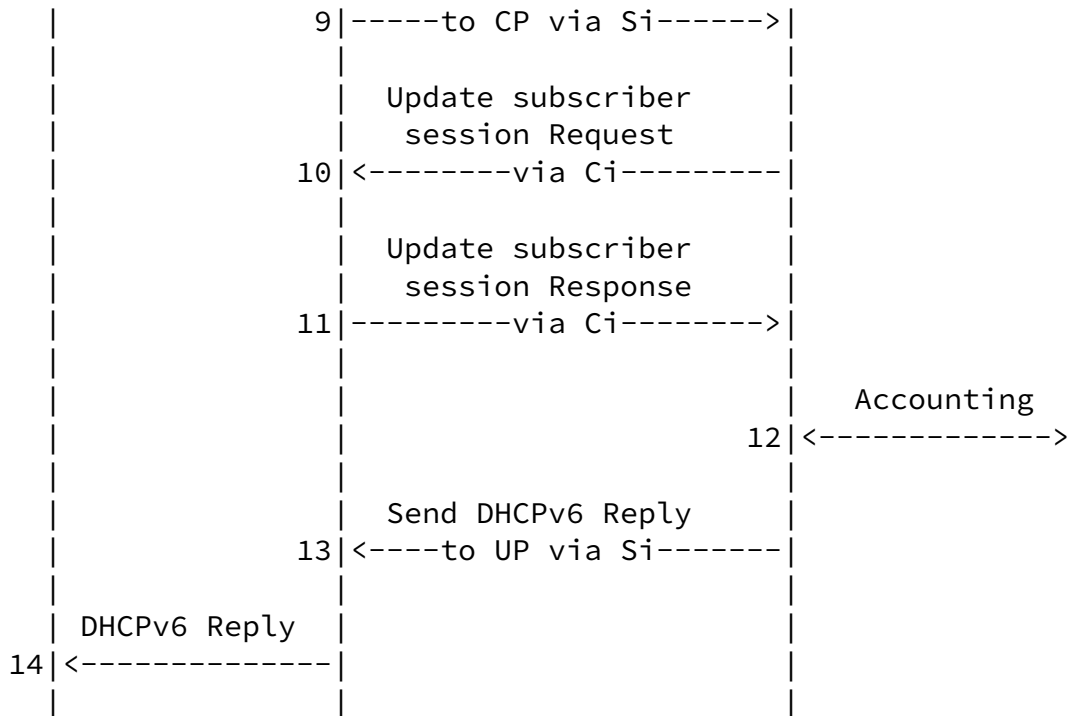


Figure 17: DHCPv6 + SLAAC Access

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:

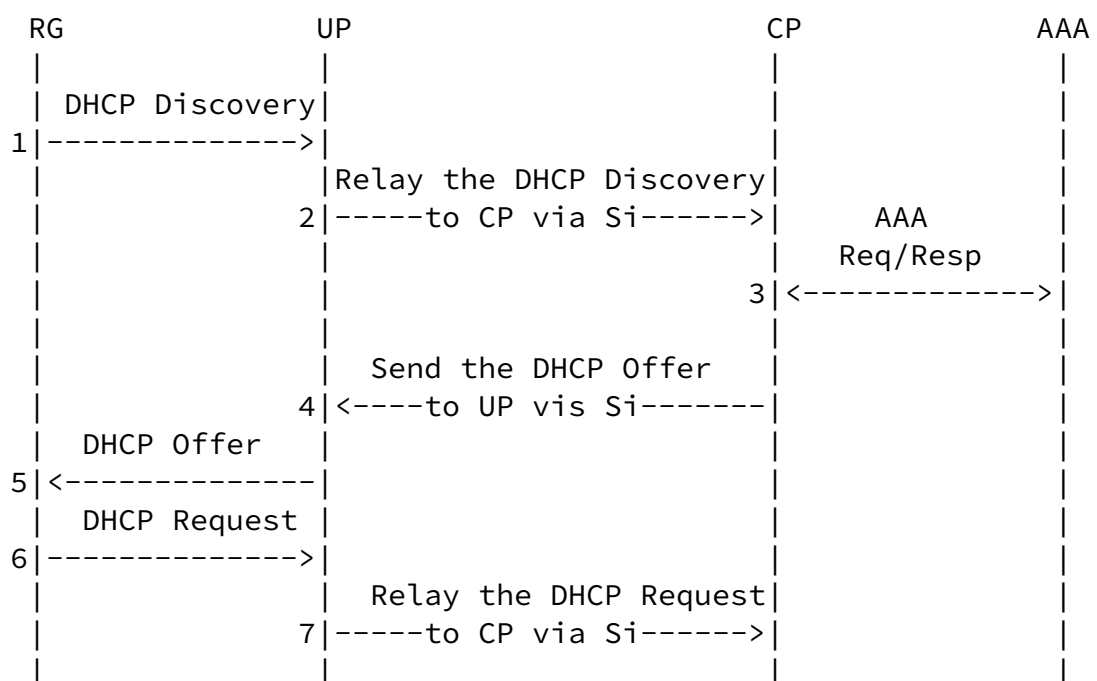
```
<Update_Request Message> ::= <Common Header>
                               <Basic Subscriber TLV>
                               <IPv6 Subscriber TLV>
                               <IPv6 Routing TLV>
                               [<Subscriber Policy TLV>]
```

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

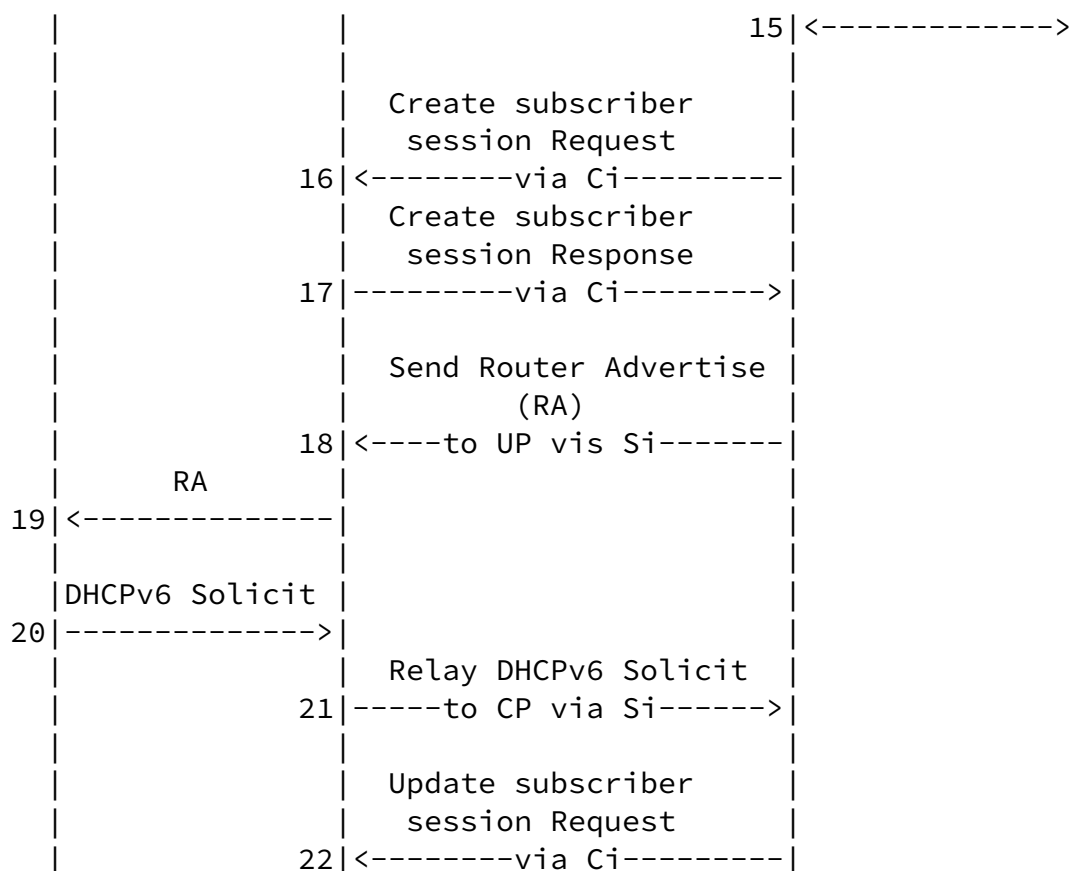
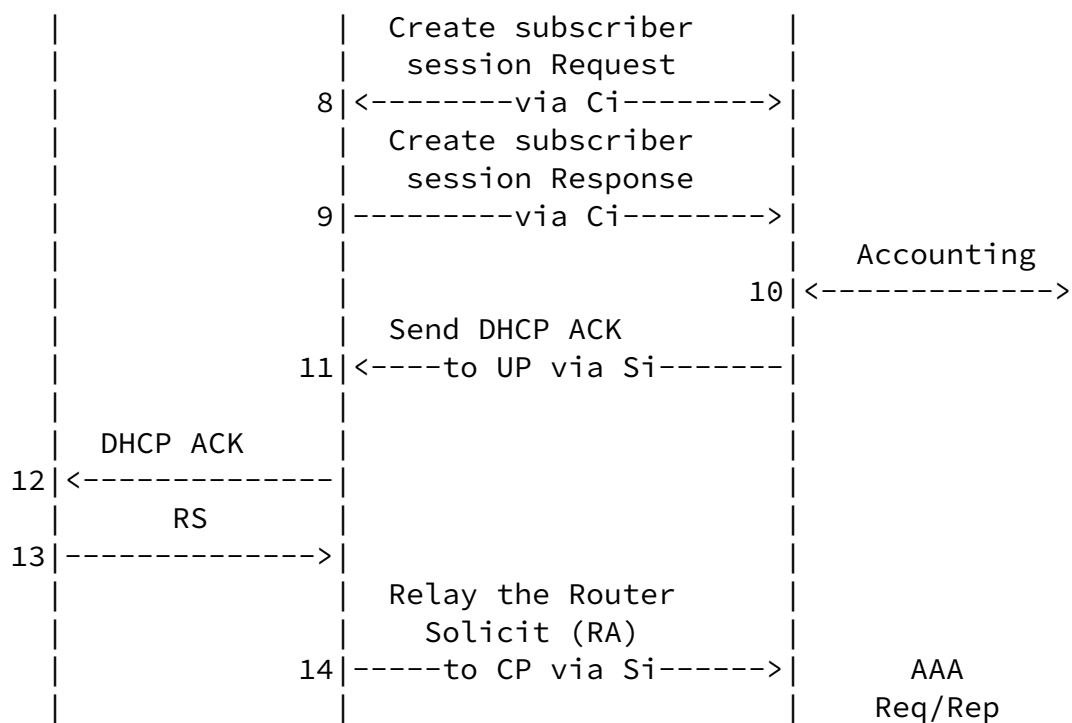
```
<Update_Response Message> ::= <Common Header>
                               <Update Response TLV>
```

#### [5.1.5](#) DHCP Dual Stack Access

The following sequence is a combination of DHCP IPv4 and DHCP IPv6 access processes.







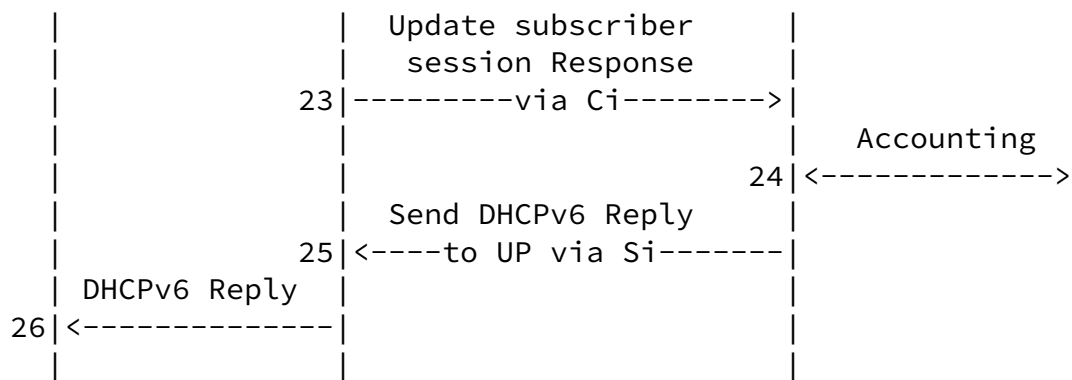


Figure 18: DHCP Dual Stack Access

The DHCP dual stack access includes three sets of Update\_Request / Update\_Response exchanges to create/update DHCPv4/v6 subscriber session.

#### 1. Create DHCPv4 session (step 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 DHCPv6 session (step 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>
  
```

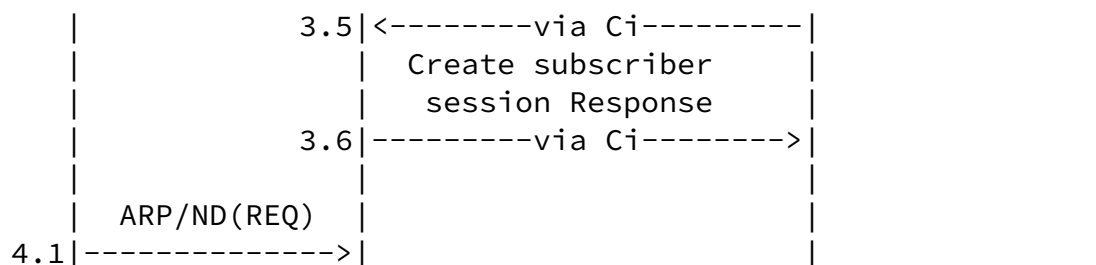
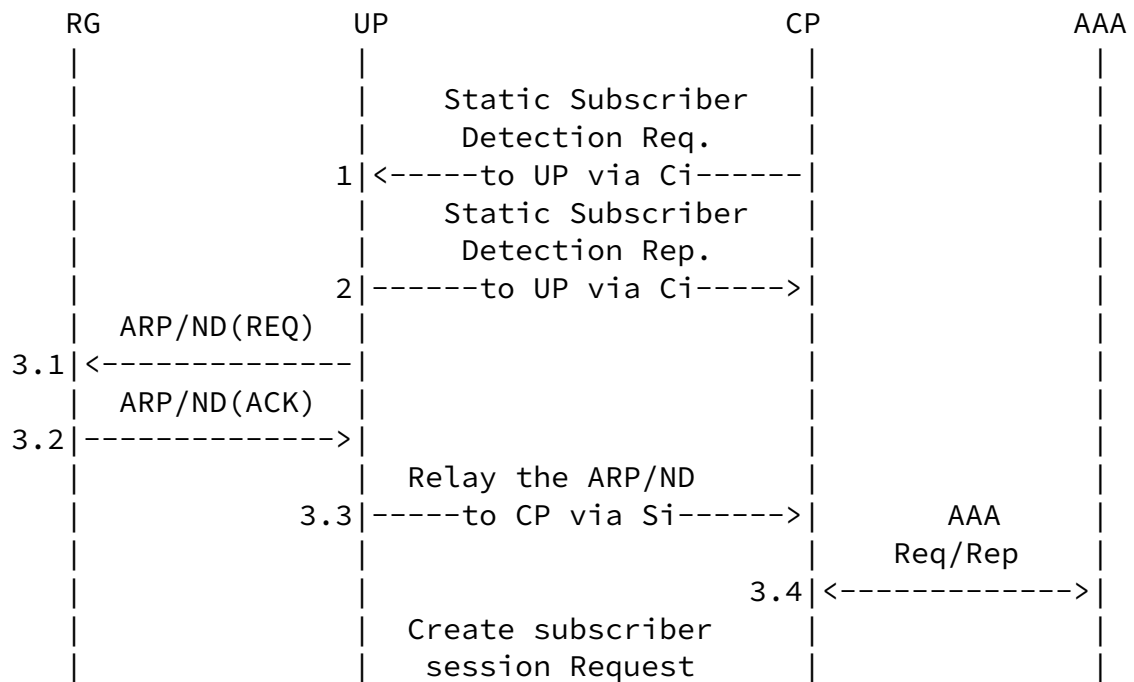
### 3. Update DHCPv6 session (step 22 and 23)

<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:



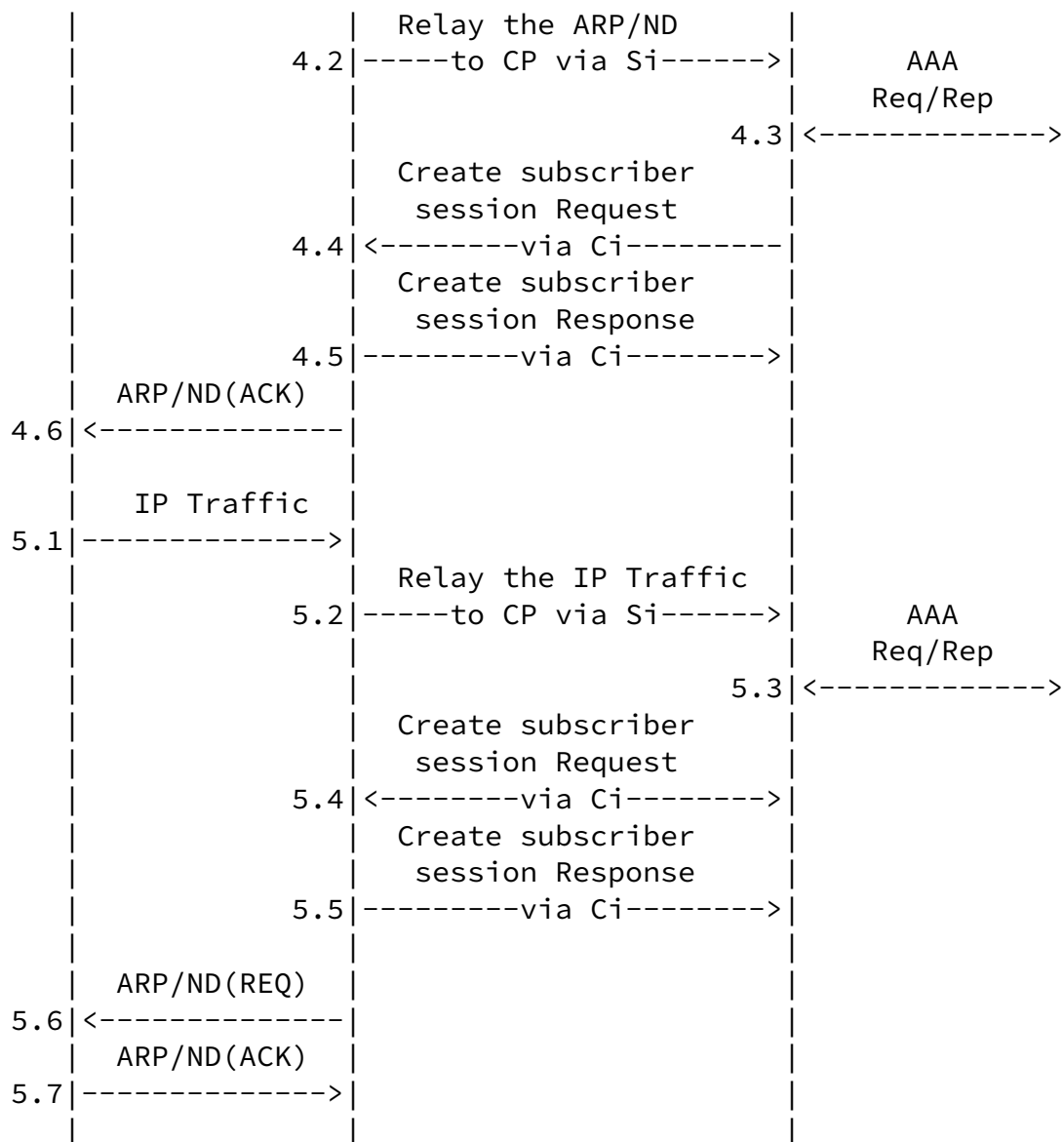


Figure 19: L2 Static Subscriber Access

For L2 static subscriber access, the process starts with a CP installing a static subscriber detection list on an UP. The list determines which subscribers will be detected. This is implemented by exchanging Update\_Request and Update\_Response messages between CP and UP. The format of the messages are as follows:



<Update\_Response Message> ::= <Common Header>  
                                  <Update Response TLV>

5.2 PPPoE

5.2.1 IPv4 PPPoE Access

The following figure shows the IPv4 PPPoE access call flow.

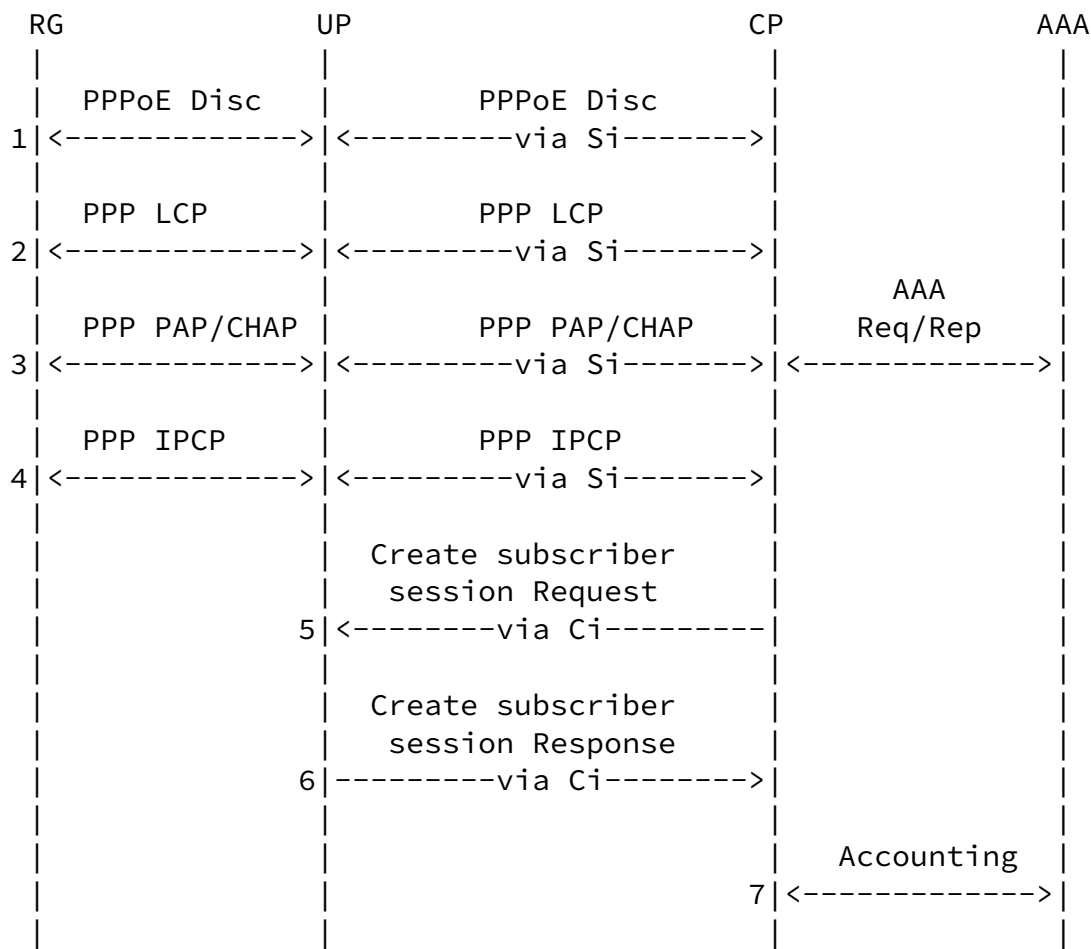


Figure 20: IPv4 PPPoE Access

From the above sequence, step 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.

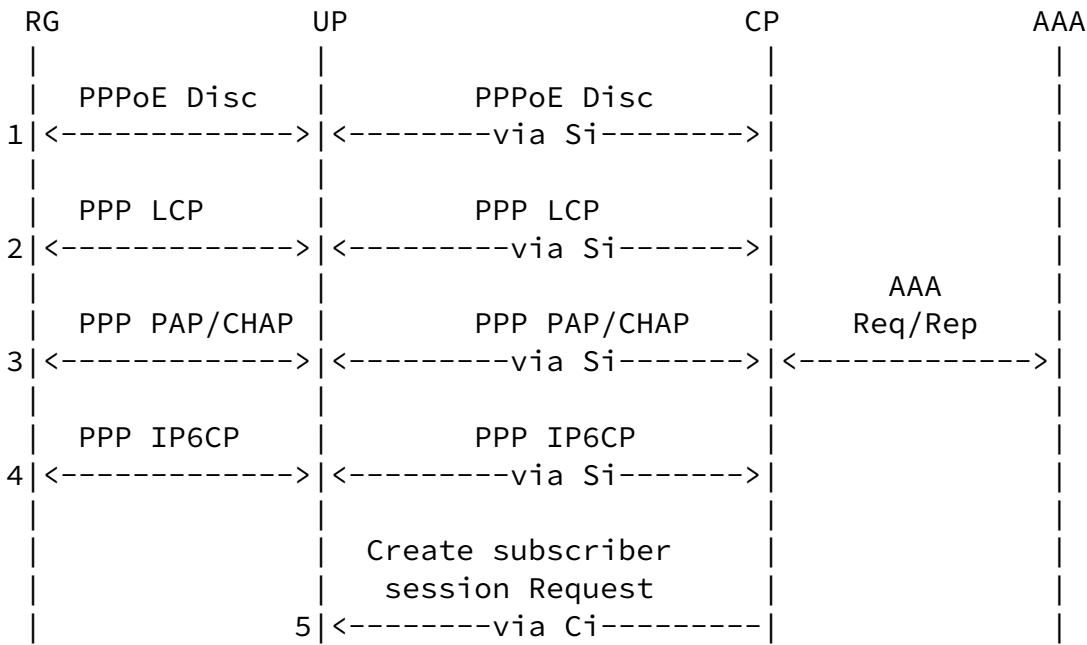
After the PPPoE call flow, if the subscriber passed the AAA authentication and authorization, the CP will create a corresponding session on the UP through the Ci. The formats of the messages are as follows:

<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

The following figure describes the IPv6 PPPoE access call flow.



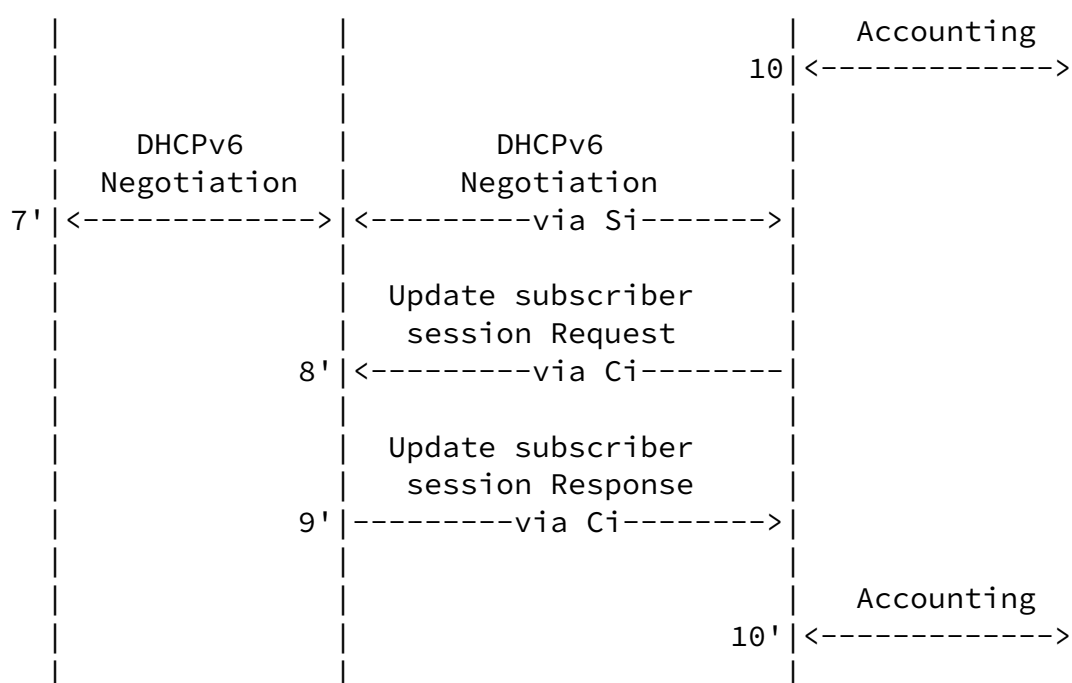
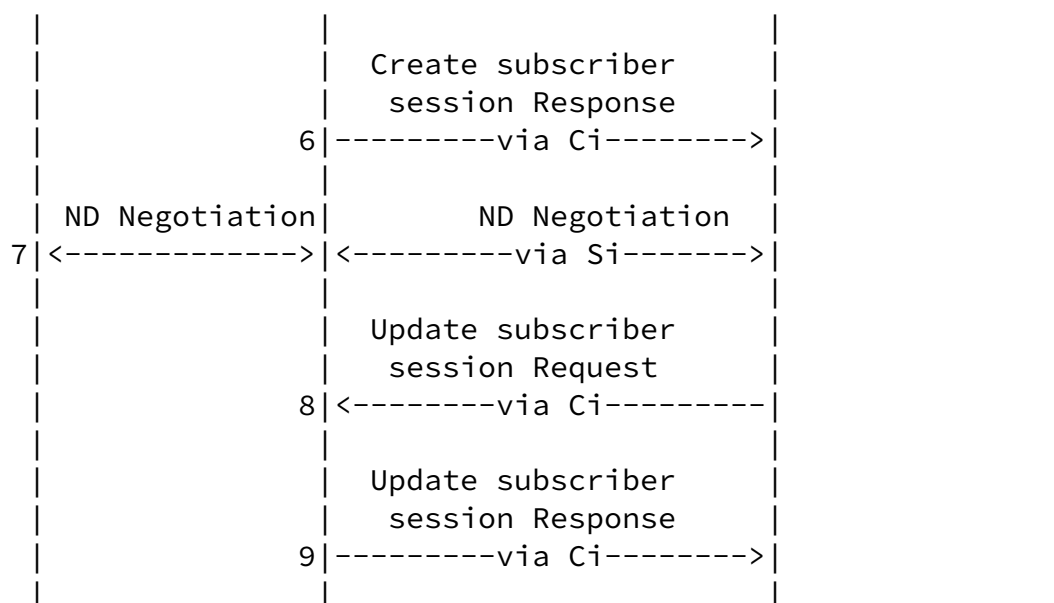


Figure 21: IPv6 PPPoE Access

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 a ND/DHCPv6 negotiation process with the CP (see step 7 and 7'), after that, it will trigger an update (8-9, 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

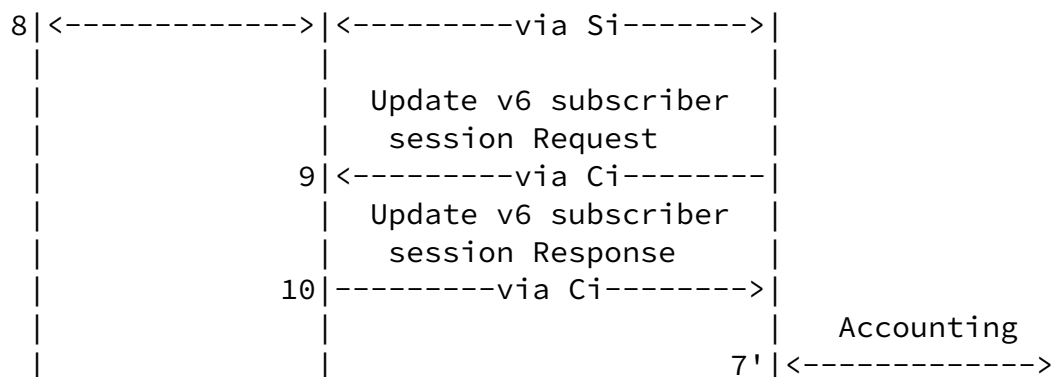
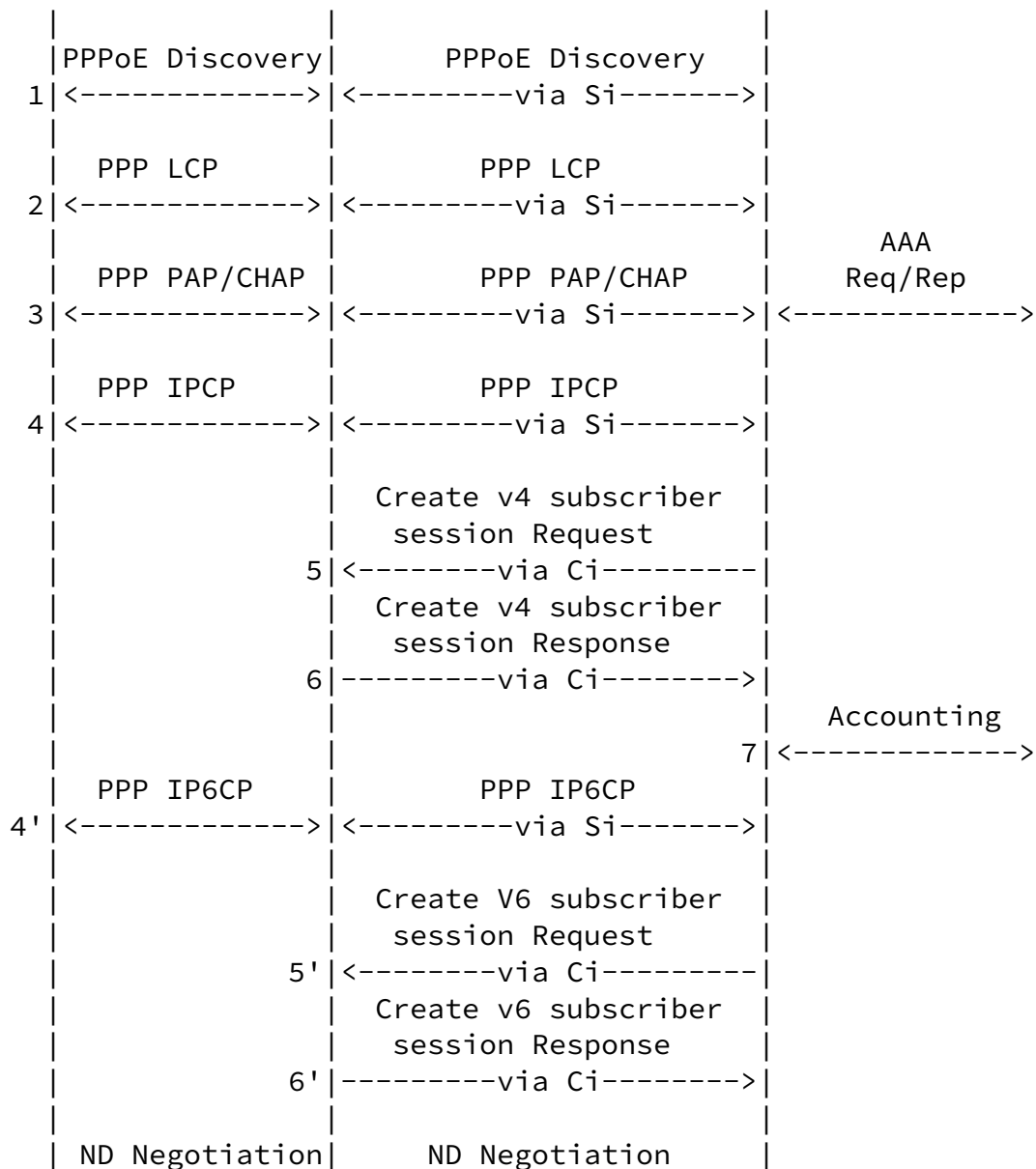
The following figure shows a combination of IPv4 and IPv6 PPPoE access call flow.

RG

UP

CP

AAA



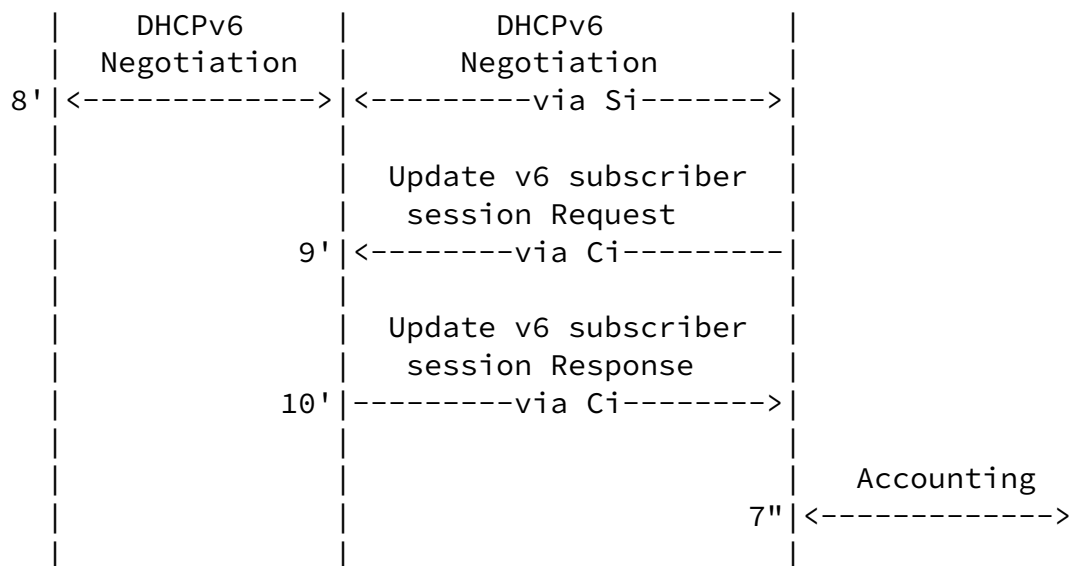


Figure 22: PPPoE Dual Stack Access

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 (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 (5'-6')

```

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

[<Subscriber Policy TLV>]

<Update\_Response Message> ::= <Common Header>  
<Update Response TLV>

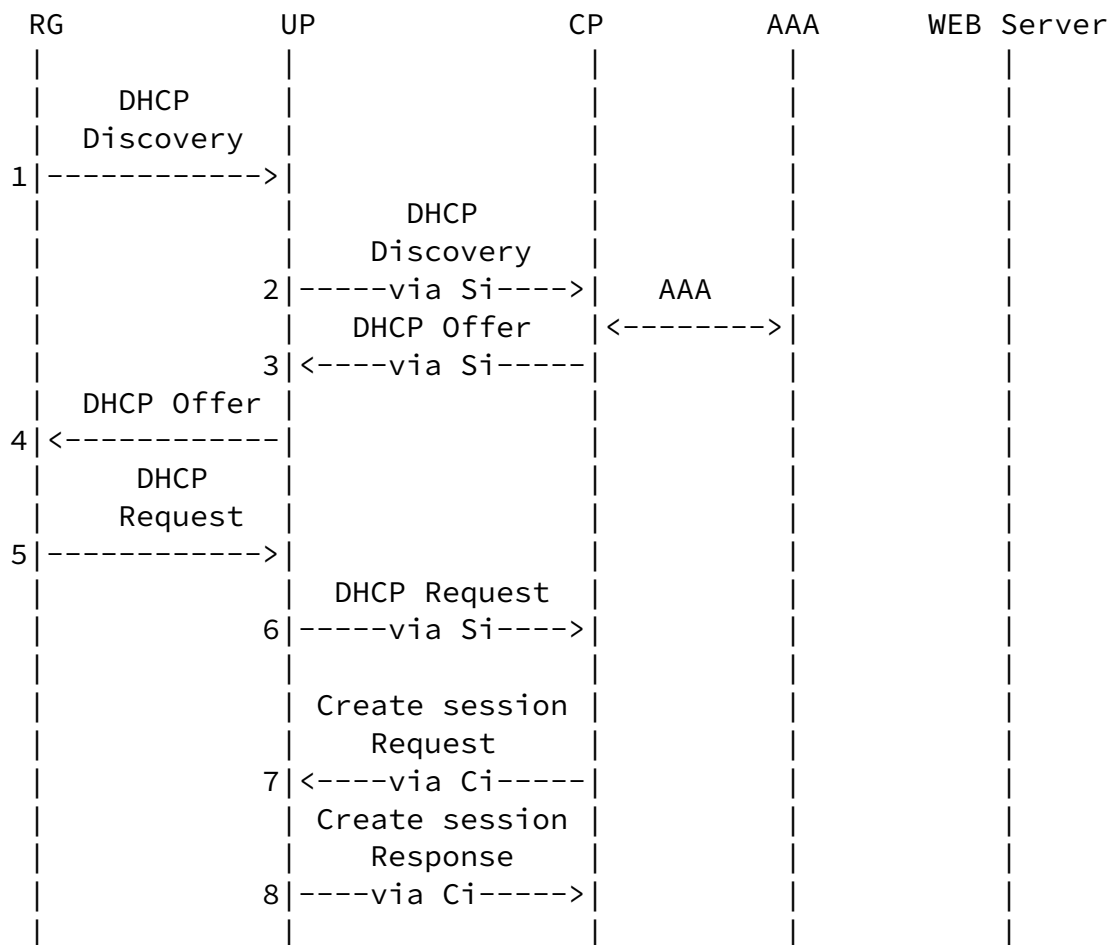
3. Update the IPv6 PPPoE subscriber session (9-10, 9'-10')

<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

The following figure shows the WLAN access call flow.





<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>]

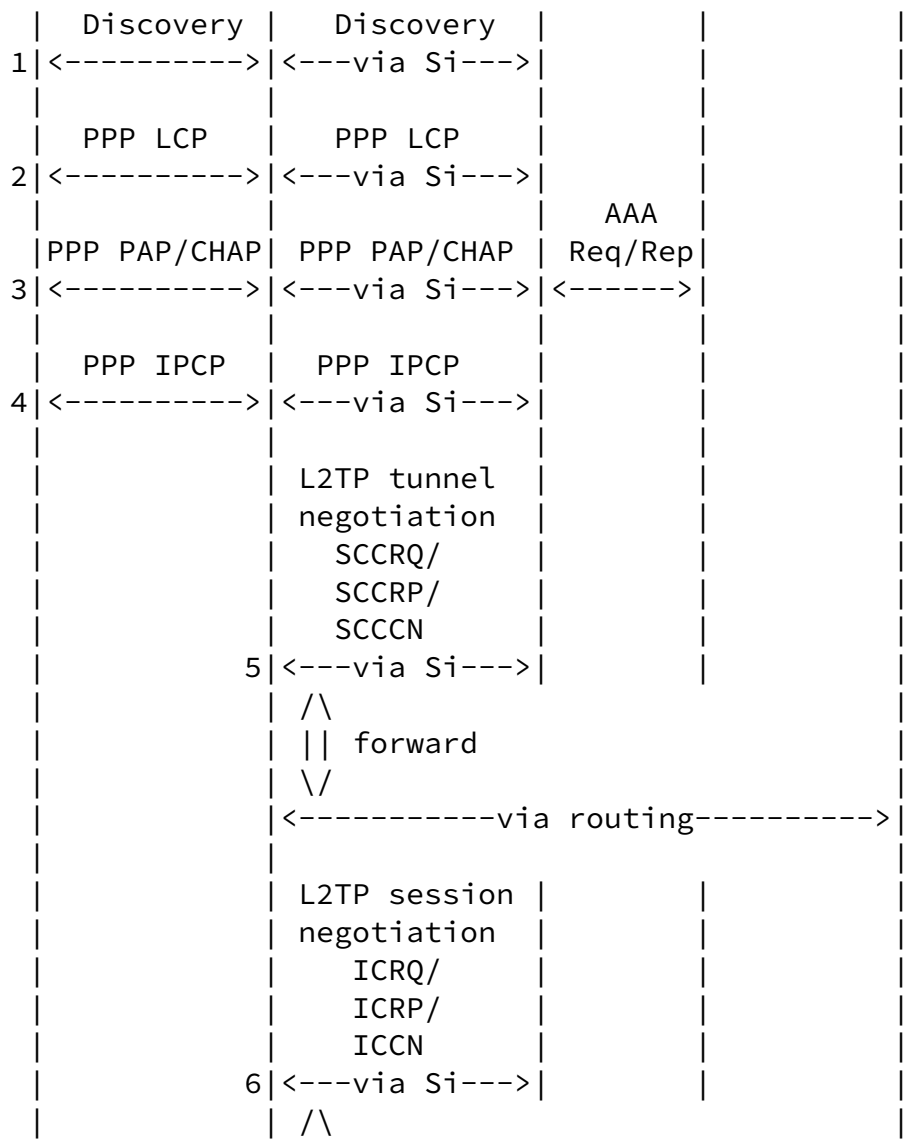
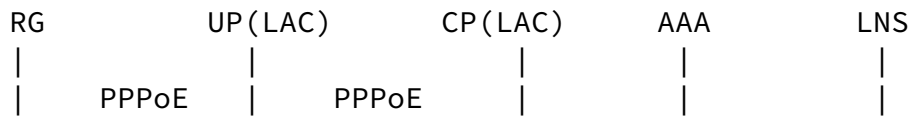
<Update\_Response Message> ::= <Common Header>  
    <Update Response TLV>  
    [<Subscriber CGN Port Range TLV>]

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



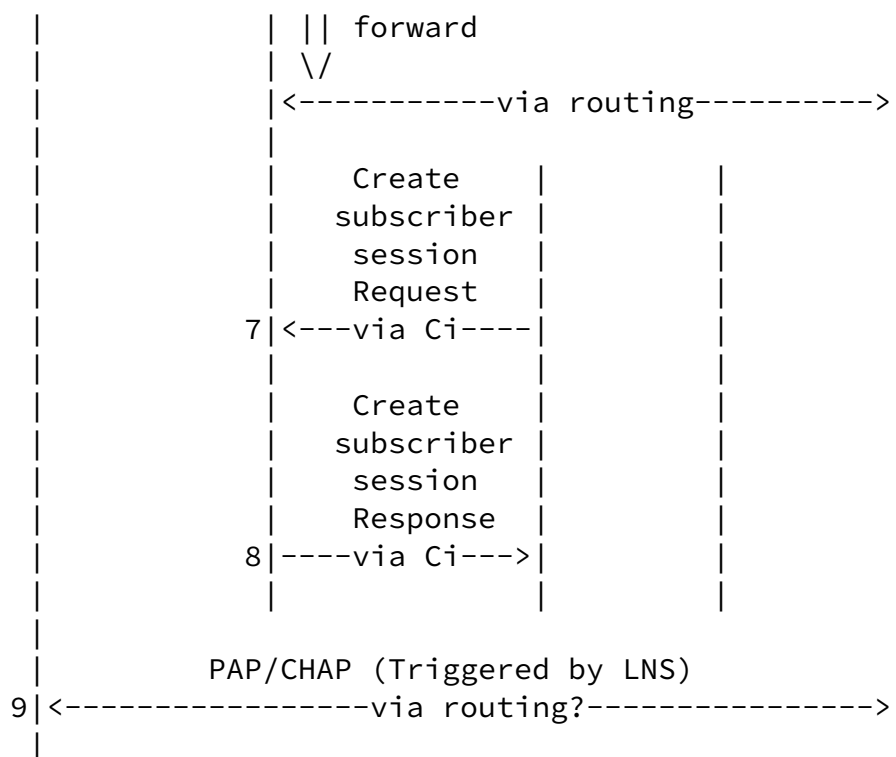


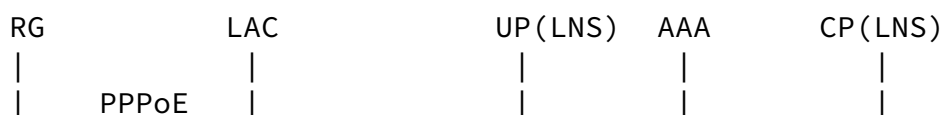
Figure 24: L2TP-LAC Access

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:

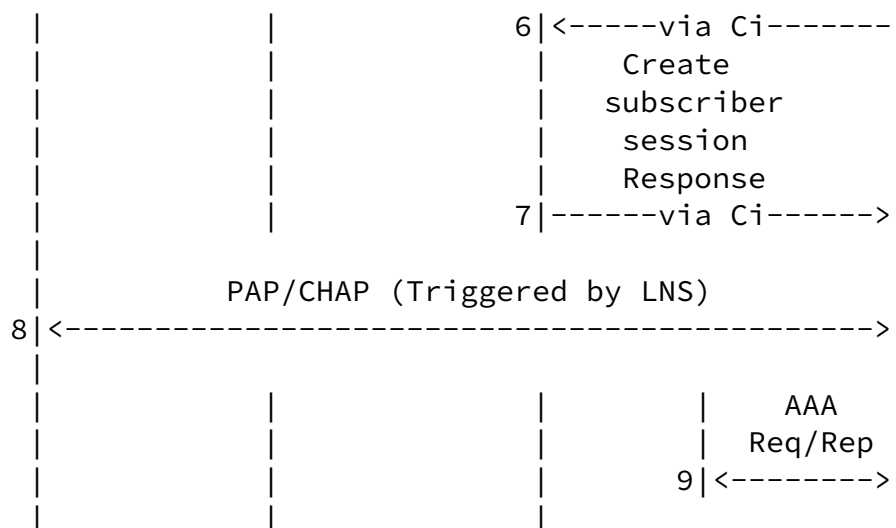
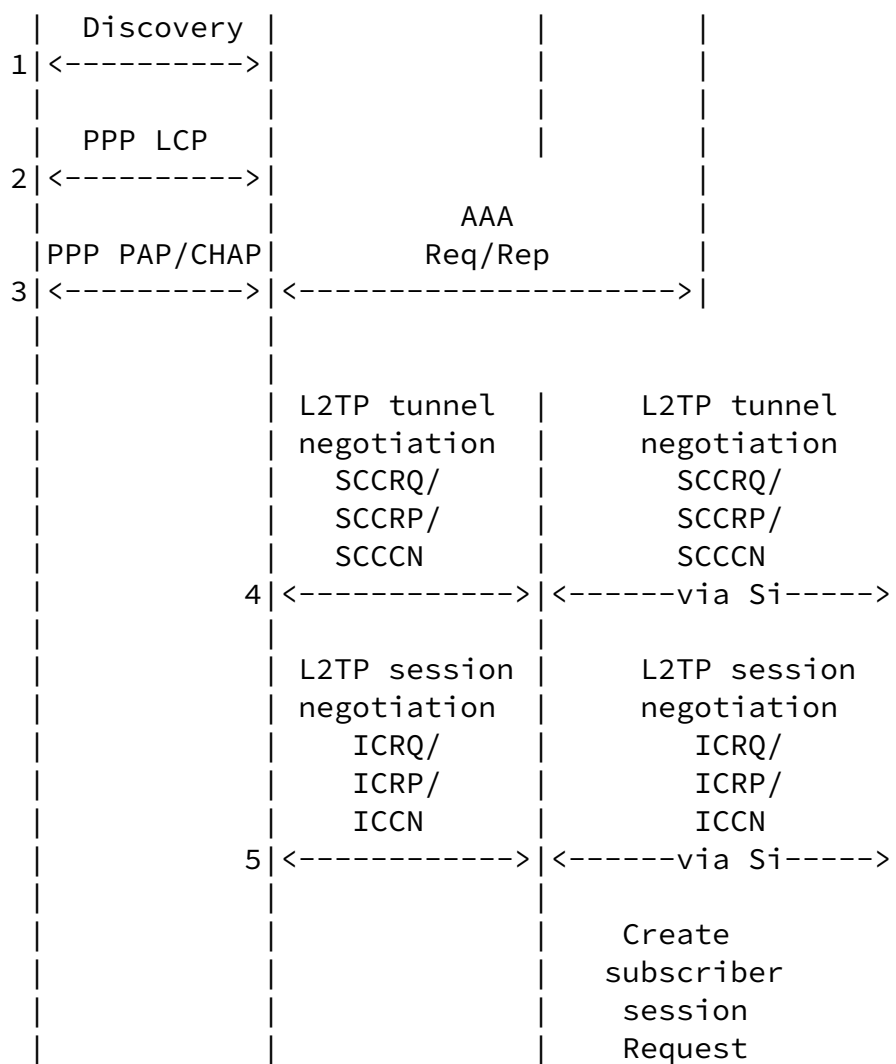
<Update\_Request Message> ::= <Common Header>  
 <Basic Subscriber TLV>  
 <L2TP-LAC Subscriber TLV>  
 <L2TP-LAC Tunnel TLV>

<Update\_Response Message> ::= <Common Header>  
 <Update Response TLV>

#### 5.4.2 L2TP LNS IPv4 Access







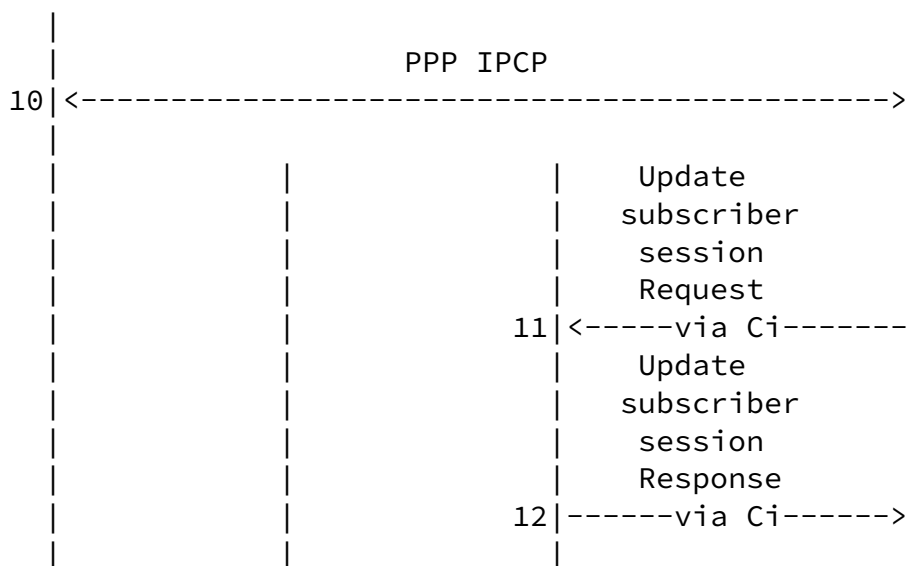


Figure 25: IPv4 L2TP-LNS Access

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 messages are 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>
```

[illegible]

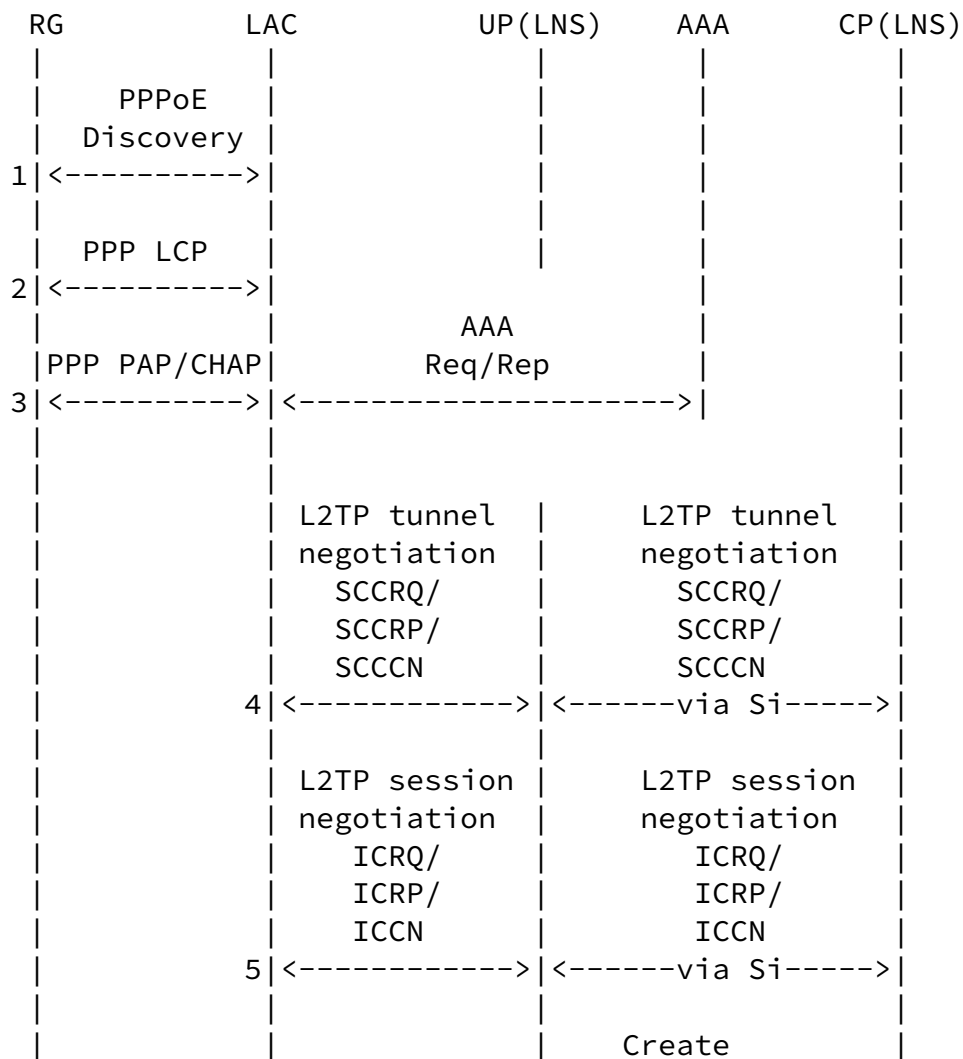
After that, the LNS-CP will trigger an AAA authentication. If the authentication result is positive, a PPP IPCP process will follow, then the CP will update the session with the following message exchanges:

```
<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



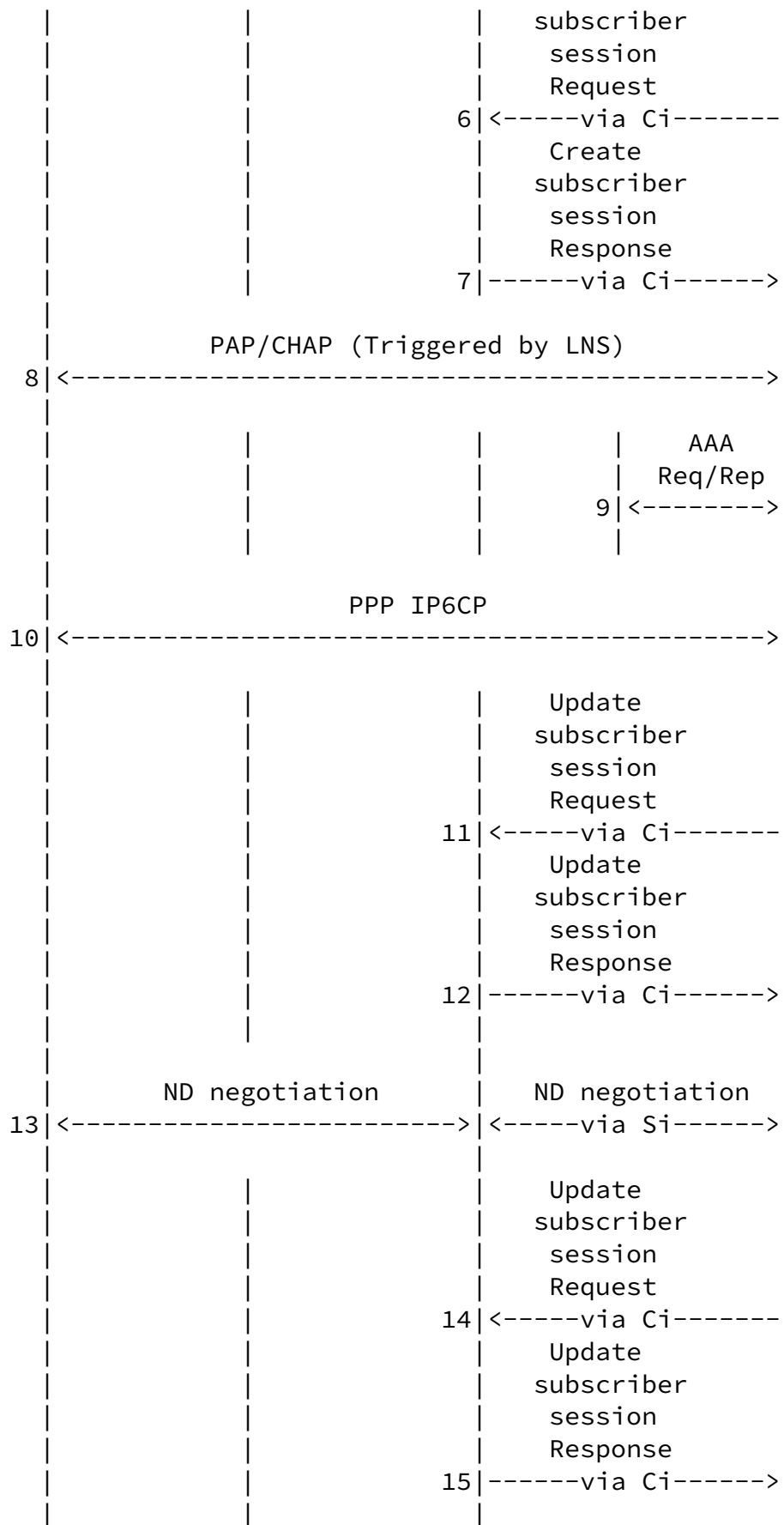


Figure 26: L2TP-LNS IPv6 Access

INTERNET-DRAFT

## Simple BNG CUSP

Steps 1-12 are the same as L2TP and 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:

[illegible]

```
<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, 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:

[illegible]

<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)

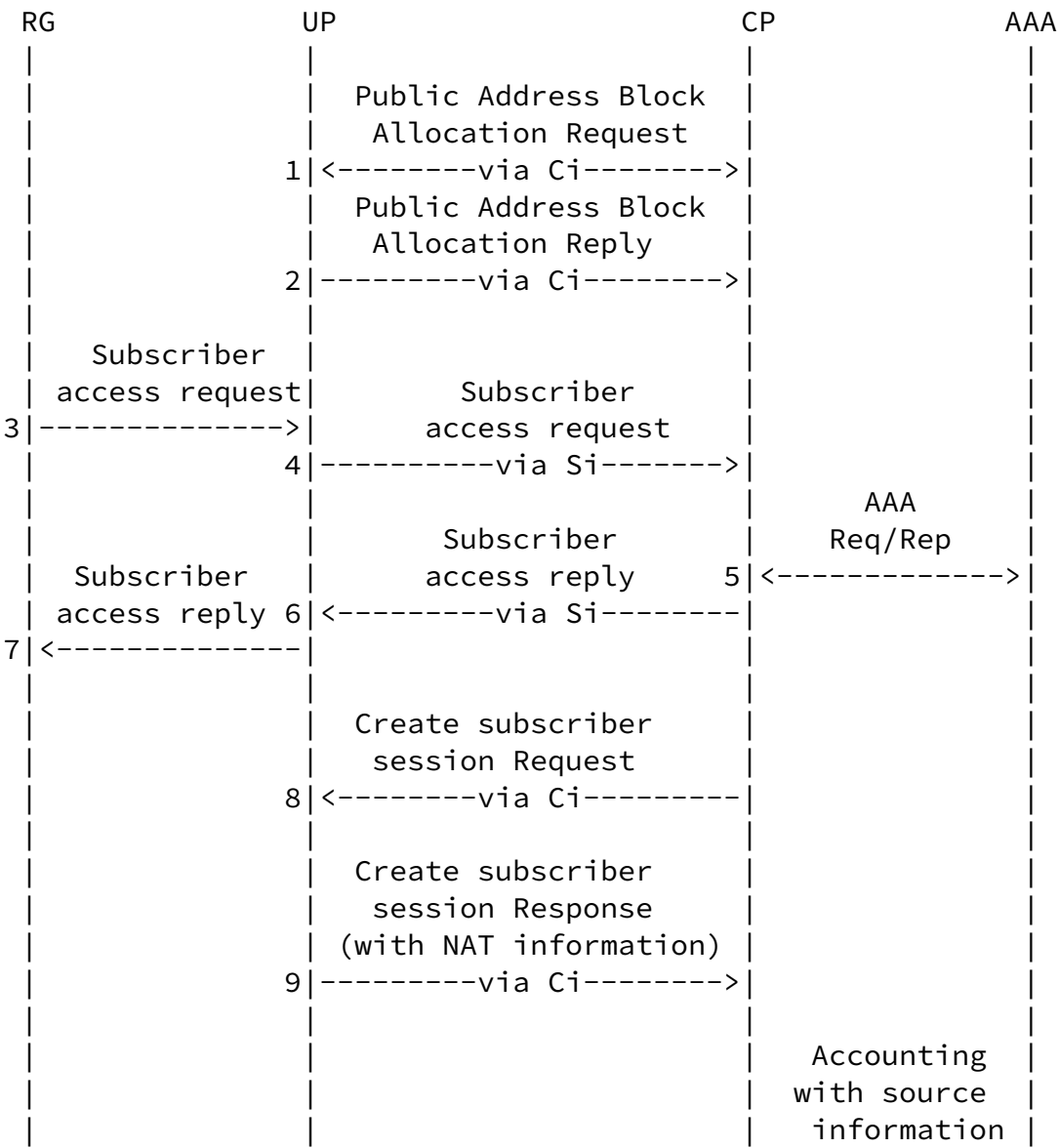




Figure 27: CGN Access

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>  
<Request Address Allocation TLV>

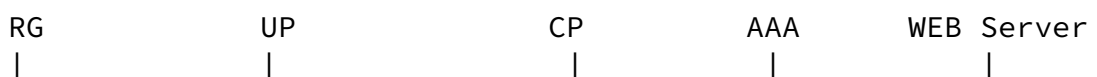
<Addr\_Allocation\_Ack Message> ::= <Common Header>  
<Address Assignment Response TLV>

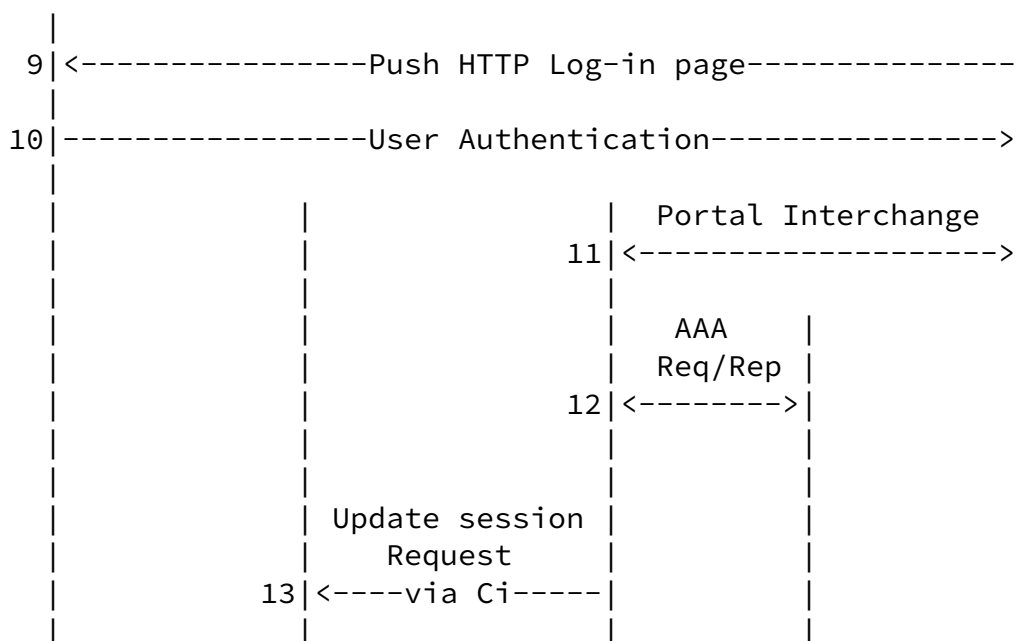
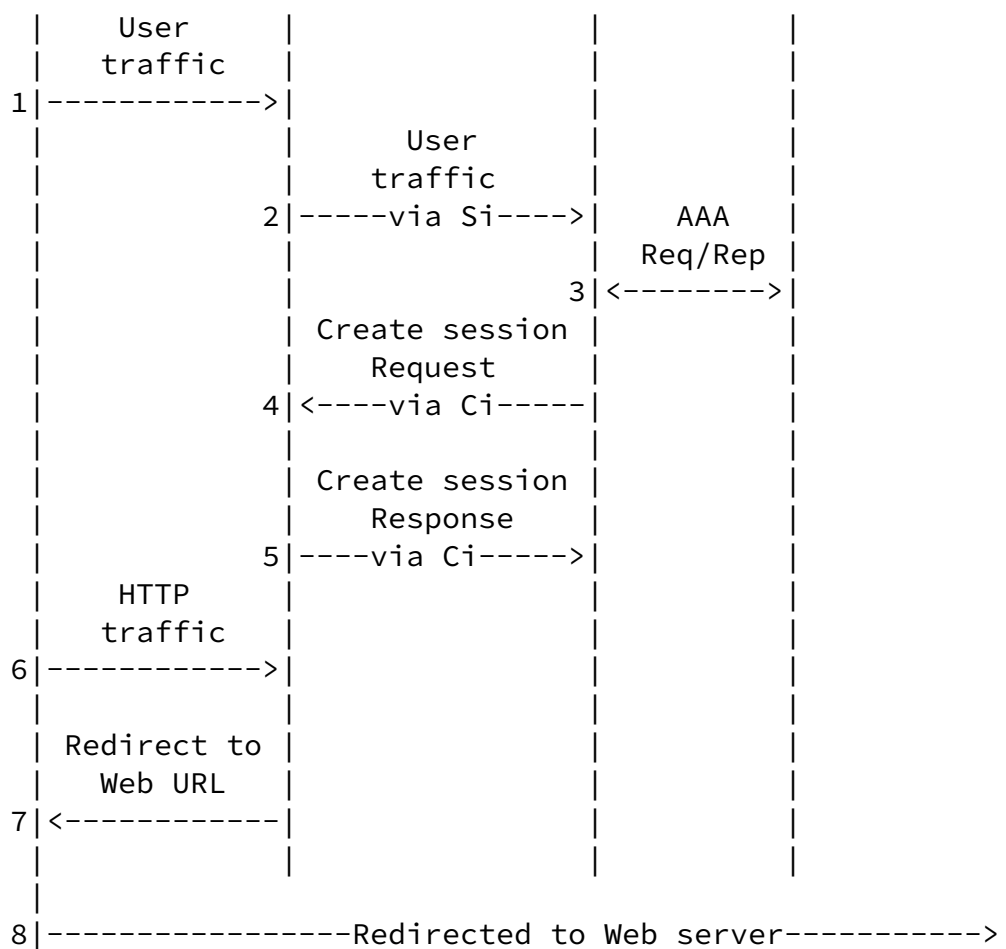
Steps 3-9 show the general dial-up process in the case of CGN mode. The specific processes (e.g., IPoE, PPPoE, L2TP, etc.) are defined in above sections.

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









```

<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>]

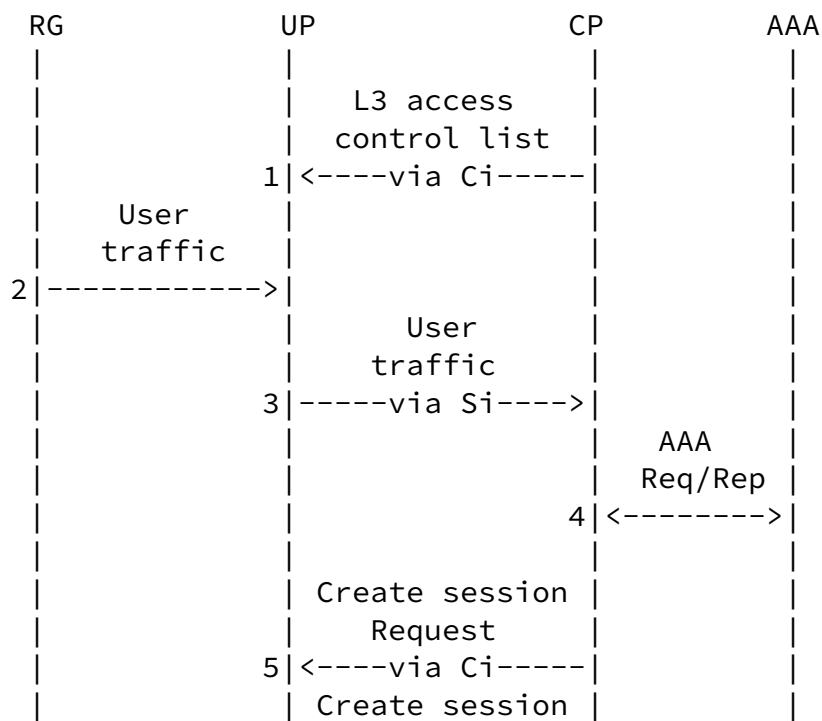
```

```

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

```

### 5.6.2 User Traffic Trigger



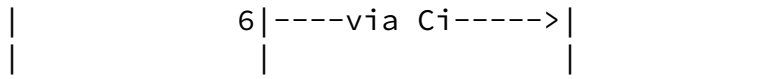


Figure 29: User Traffic Triggered L3 Leased Line Access

In this user traffic triggered case, the CP must install an access control list on the UP, which is used by the UP to determine whether an RG is legal or not. 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:

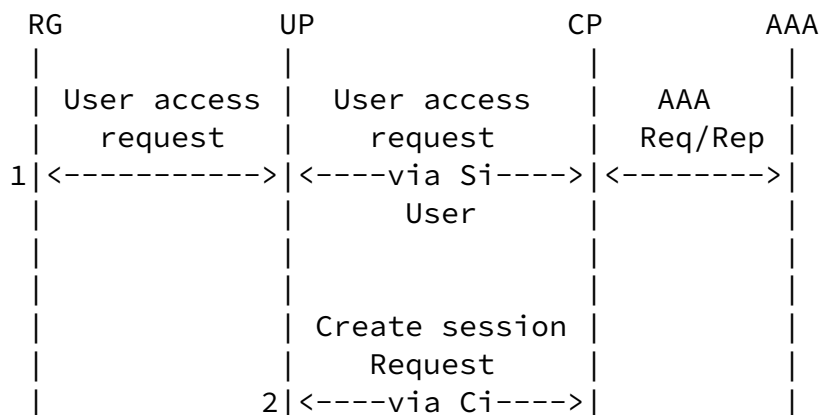
```

<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 Access



INTERNET-DRAFT

Simple BNG CUSP

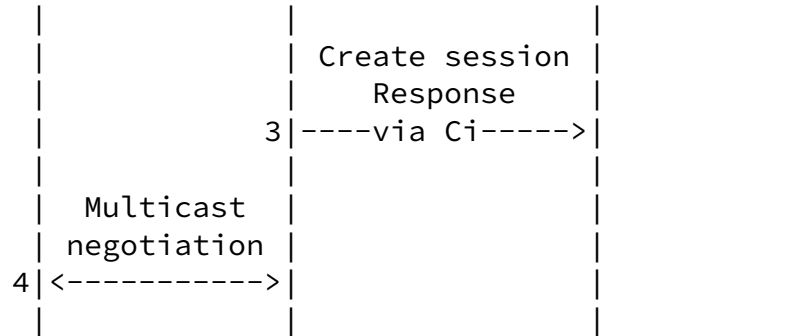


Figure 30: Multicast Access

Multicast access starts with an 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:

```

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

```

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

INTERNET-DRAFT

Simple BNG CUSP

## 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

S-CUSP Common Message Header:

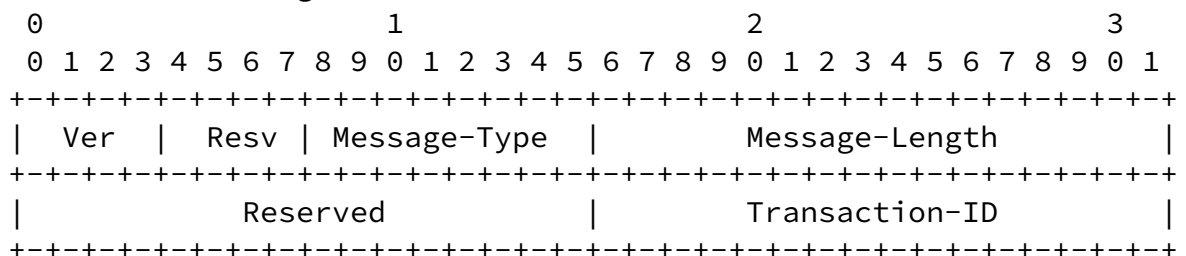


Figure 6.1: S-CUSP Message Common Header

- o 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 structure and format 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.

- o Resv (4 bits): Reserved. MUST be sent as zero and ignored on receipt.
- o Message-Type (8 bits): The set of message types specified in this document is listed in [Section 9.1](#).
- o Message-Length (16 bits): Total length of the S-CUSP message including the common header, expressed in number of bytes as an unsigned integer.

- o 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 message and that value wrap back to zero after 0xFFFF. The contents 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, Keep-Alive 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 <a href="#">Section 5</a> .
6	Sync_End	End synchronization.
7	Update_Request	TLVs specified in Sections <a href="#">7.6-7.9</a> .
8	Update_Response	TLVs specified in Sections <a href="#">7.6-7.9</a> .

### [6.2.1](#) Hello Message

Hello message is used for S-CUSP session establishment and version negotiation. The detail of S-CUSP session establishment and version negotiation can be found in [Section 4.1.1](#).

The format of 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: One or more of the TLVs was not understood.

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

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

1003: The establishment timer expires. session establishment phase fails.

### [6.2.2](#) Keepalive Message

The Keepalive message is periodically sent by each end of an S-CUSP session. It is used to detect whether the peer end is still alive. The Keepalive procedures are defined [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 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: 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>
                        [<Resource Reporting TLVs>]
```

Synchronization from CP to UP: Synchronize all subscriber sessions to UP. As for which TLVs should be carried, it depends on the specific session data to be synchronized. This is equivalent to create the specific session. Refer to [Section 5](#) to see more details.

```
<Sync_Data Message> ::= <Common Header>
                        [<User Routing TLVs>]
                        [<User Information TLVs>]
                        [<L2TP Subscriber TLVs>]
                        [<Subscriber CGN Port Range TLV>]
                        [<Subscriber Policy TLV>]
```

#### [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: One or more of the TLVs was not understood.

#### [6.2.7](#) Update\_Request Message

The Update\_Request message is a multi-task 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" can be set to either "update" or "delete" when a TLV is carried in an Update\_Request message.

When the "Oper" set to update, it means to create or update a subscriber session, if the "Oper" set to delete, it indicates to delete a corresponding session on an UP.

The format of Update\_Request message is as follows:

```
<Update_Request Message> ::= <Common Header>
                               [<User Routing TLVs>]
                               [<User Information TLVs>]
                               [<L2TP Subscriber TLVs>]
                               [<Subscriber CGN Port Range TLV>]
                               [<Subscriber Policy TLV>]
```

Each Update\_Request message will result in an Update\_Response message that 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: 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 Event message is as follows:

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

## [6.4](#) Report Message

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

### [6.5.1](#) Addr\_Allocation\_Req Message

The Addr\_Allocation\_Req message is used to request CGN address allocation. The format of 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 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 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 Addr\_Renew\_Ack 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 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 Addr\_Release\_Ack message is as follows:

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

## [6.6](#) Vendor Message

The Vendor message is, in conjunction with the vendor TLV and vendor sub-TLV, can be used by vendors to extend the S-CUSP protocol. It's 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 know the message type of a 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:

- o the format of the TLVs that appear in S-CUSP messages,
- o the format of the sub-TLVs that appear within the values of some TLVs, and
- o the format of some basic data fields that appear within TLVs or sub-TLVs.

See [Section 9](#) 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.

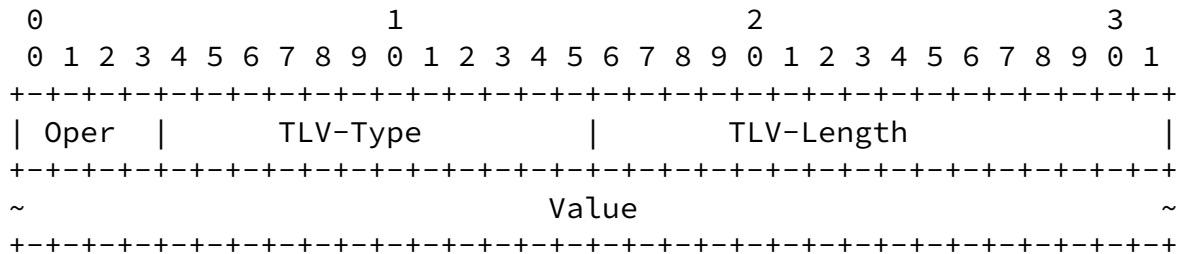


Figure 32: Common TLV Header

- o Oper (4 bits): For Message-Types that indicate an operation on a data set, the Oper field is interpreted as Update, Delete, or Reserved as specified in [Section 9.3](#). For all other Message-Types, the Oper field MUST be sent as zero and ignored on receipt.
- o TLV-Type (12 bits): The Type of a TLV, that is the meaning and format of the Value part, are determined by the TLV-Type of the TLV. See [Section 9.2](#).
- o TLV-Length (2 bytes): The length of the Value portion of the TLV in bytes as an unsigned integer.
- o Value (variable length): This is the value portion of the TLV whose size is given by TLV-Length. The value portion 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.

- o 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. These labels will not normally be seen by subscribers. They are normally interpreted as ASCII [[RFC20](#)].

- o MAC-Addr: 6 octets. Ethernet MAC Address [[RFC7042](#)].
- o 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.
- o IPv6-Address: 20 octets. 16 octets of 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  $2^{*(128-n)} - 1$ .
- o VLAN ID: 2 octets. As follows [[802.1Q](#)]:

```

      0                               1
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
| PRI |D|          VLAN-ID          |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+

```

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 above, can contain one or more Sub-TLVs formatted as follows:

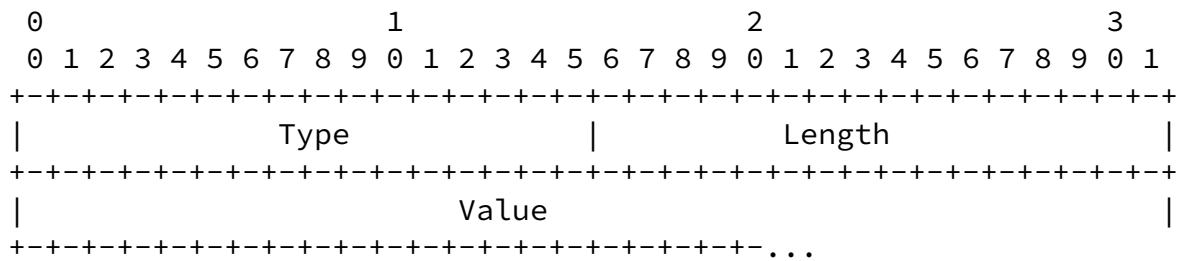


Figure 33: Sub-TLV Header

- o Type (2 bytes): The Type of a Sub-TLV, that is the meaning and format of the Value part, are determined by the Type of the TLV. Sub-TLV Types numbers have the same meaning regardless of the TLV Type of the TLV within which the sub-TLV occurs. See [Section 9.4](#).
- o Length (2 bytes): The length of the Value portion of the sub-TLV in bytes as an unsigned integer.
- o 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-TLV is variable from 1 to 255 octets. The value is of type STRING padded with zeros octets to 4-octet alignment.

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
7	NAT-Instance	The name of a NAT instance
8	Pool-Name	The name of an address pool

### 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 9 and the sub-TLV length is 16. The format is as shown in Figure 34.

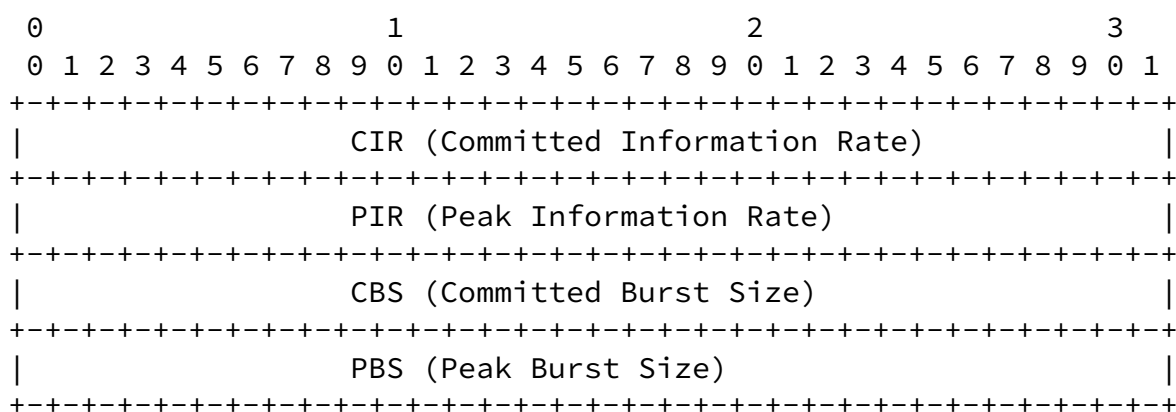


Figure 34: Ingress-CAR sub-TLV

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 10 and its sub-TLV length is 16 octets. The format of the value part is as defined below.

INTERNET-DRAFT

Simple BNG CUSP

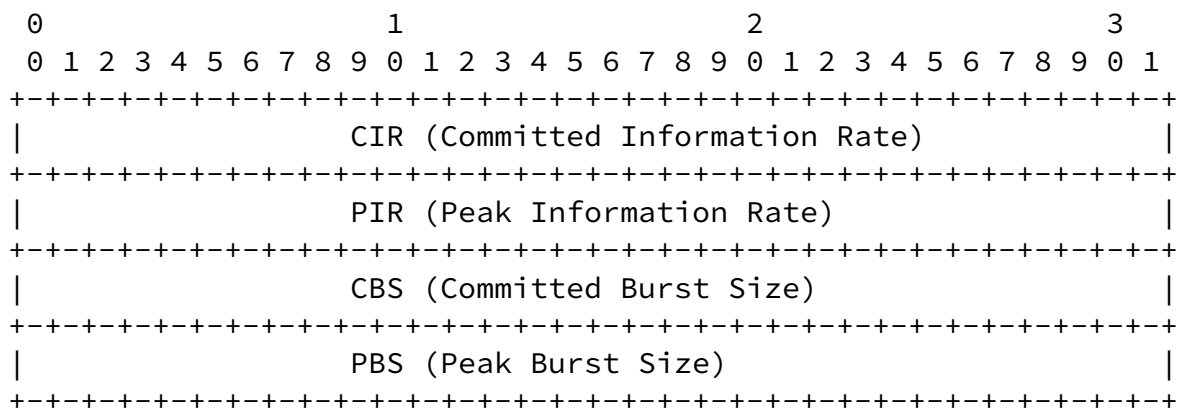


Figure 35: Egress-CAR sub-TLV

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 local unique identifier within a BNG.

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

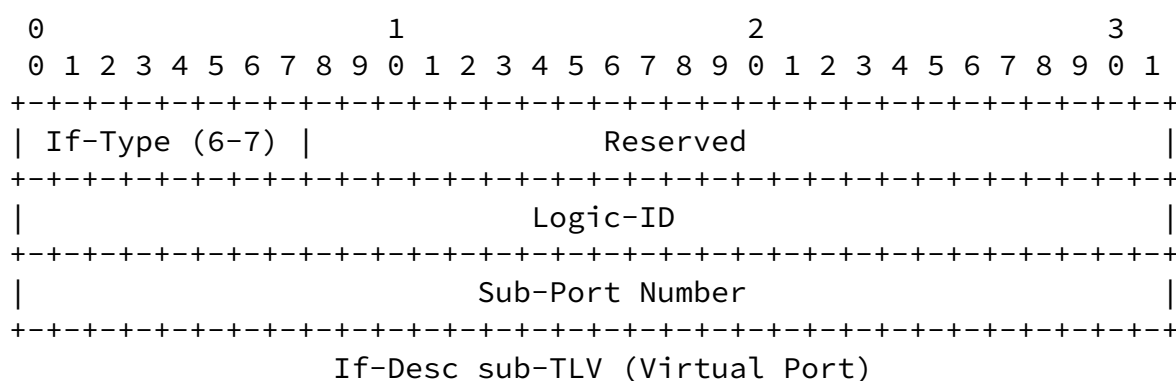
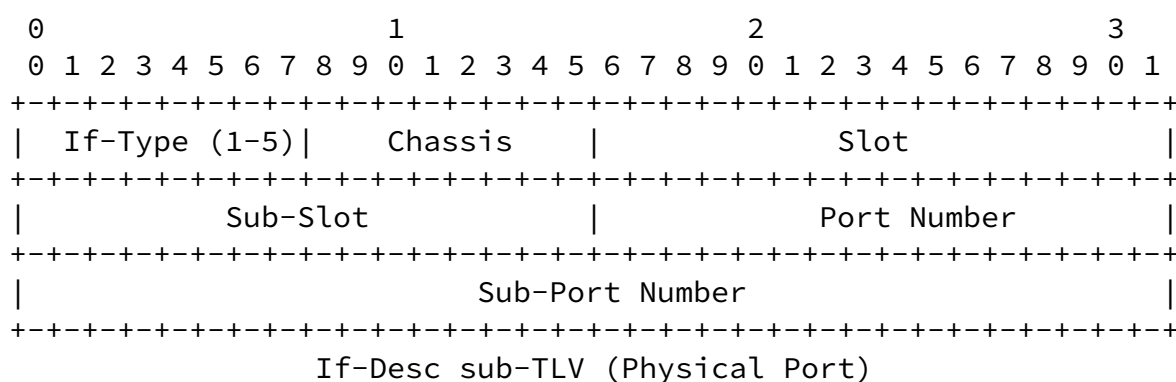


Figure 36: If-Desc sub-TLV Formats

Where:

If-Type: 8 bits in length, indicates the type of an interface.  
Following types are defined in this document:

- 0: Reserved
- 1: Fast Ethernet (FE)
- 2: GE
- 3: 10GE
- 4: 100GE
- 5: Eth-Trunk
- 6: Tunnel
- 7: VE
- 8-255: Reserved.

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 IPv6 Addresses sub-TLV is as follows:

0	1	2	3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1			

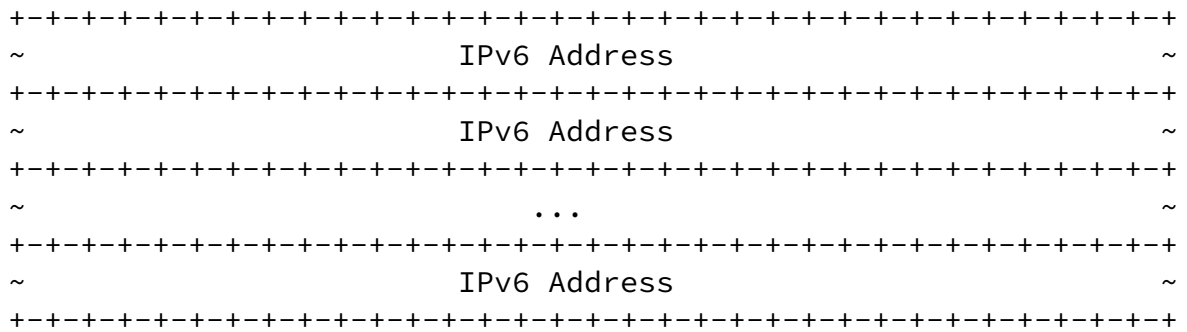


Figure 37: IPv6 Address List sub-TLV

Where:

IP Address (IPv6-Address): Each IP Address is an IP-Address type, carries an IPv6 address.

### 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:

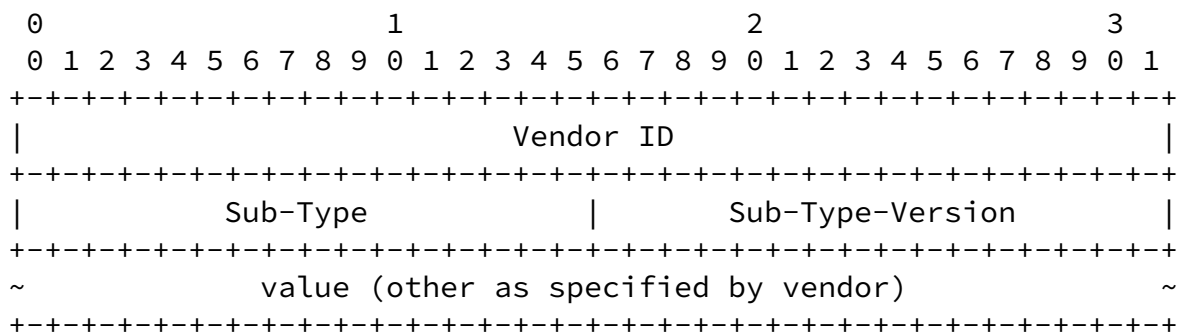


Figure 38: Vendor sub-TLV

Where:

The sub-TLV type: 13.

The 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 is 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.



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 Hello TLV is as follows:

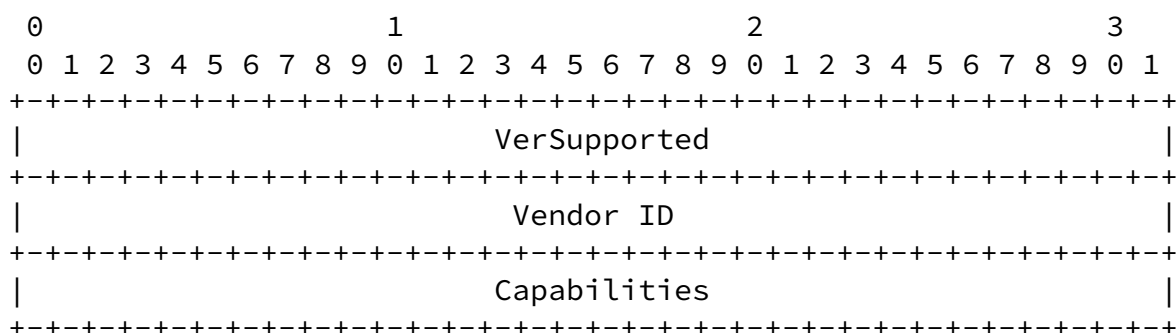


Figure 39: Hello TLV

Where:

The TLV type is 100.

The TLV length is 12 octets.

The value field consists of three parts:

**VerSupported:** 32 bits in length. This is a bit map of the Sub-Versions of the S-CUSP protocol that the sender supports. This document specifies Sub-Version zero of Major Version 1, that is, Version 1.0. The VerSupported field **MUST** be non-zero. 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 and so implementations 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 bits fields 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 non-zero, 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 non-zero, 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](#) The Keep Alive TLV

The Keep Alive TLV is defined to be carried in the Hello message. It provides timing information for the keep alive feature. The format of Hello TLV is as follows:

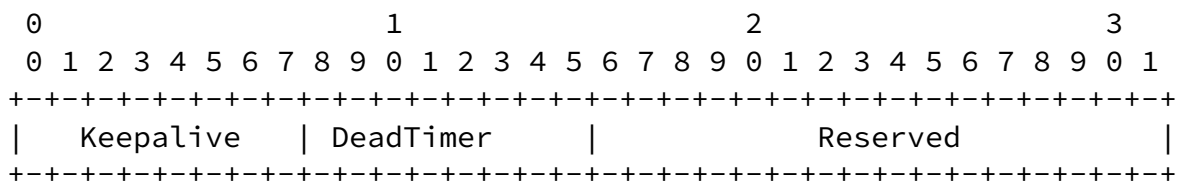


Figure 40: Keep Alive TLV

Where:

The TLV type: 102.

The value of length: 4 octets.

Keepalive (8 bits): Indicates the maximum period of time (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 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.

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

7.6 The Error Information TLV

The Error Information TLV is a common TLV that can be used in many Response (e.g., Update\_Response message) and ACK messages (e.g., Addr\_Allocation\_Ack message, etc.). It is used to convey the information about an error in the received S-CUSP message. The format of the Error Information TLV is as follows:

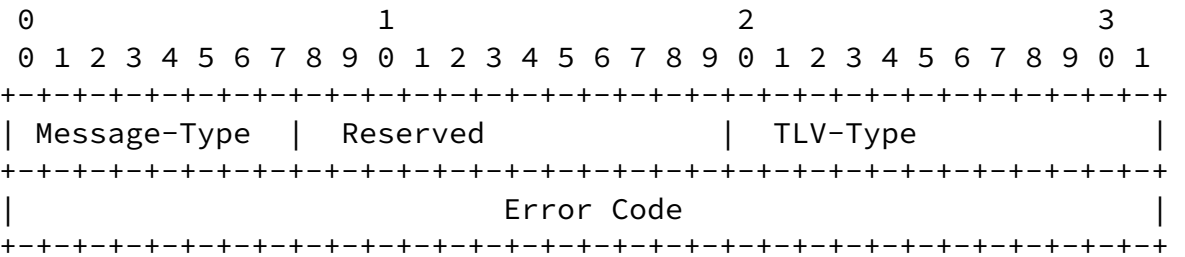


Figure 41: Error Information TLV

Where:

The TLV type: 101.

The value of 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 9.5](#)).

## 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

Hu, et al

[Page 79]

INTERNET-DRAFT

## Simple BNG CUSP

on a UP.

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

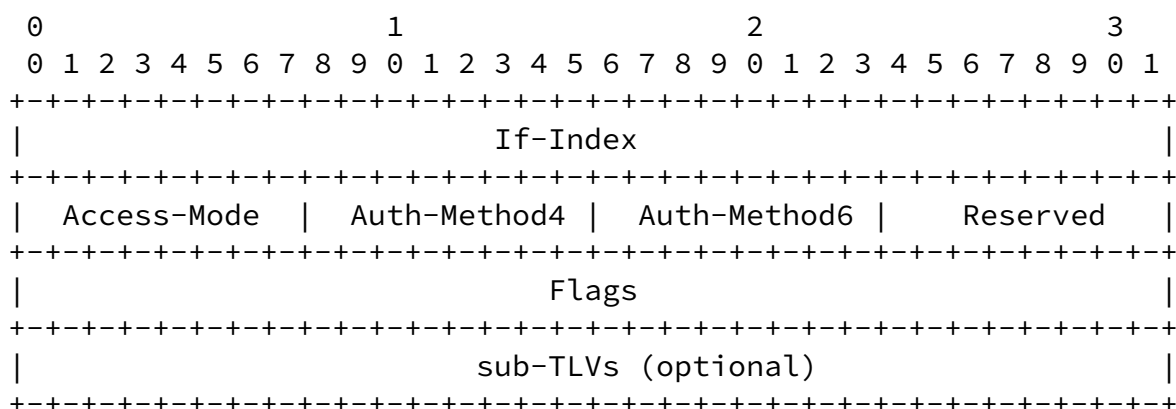


Figure 42: BAS Function TLV

Where:

The TLV type: 1.

The value of length: variable.

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

Access-Mode: 1 byte in length, indicates the access mode of the interface. This document defines following values:

0: Layer 2 subscriber;  
 1: Layer 3 subscriber;  
 2: Layer 2 leased line;  
 3: Layer 3 leased line;  
 4-255: Reserved.

Auth-Method4: 1 byte in length, for IPv4 scenario, it indicates the authentication on this interface; this field is defined as a bitmap, this document defines following values (other bits are reserved and MUST be sent as zero and ignored on receipt):

0x1: PPPoE authentication;  
 0x2: DOT1X authentication;  
 0x4: Web authentication;  
 0x8: Web fast authentication;  
 0x10: Binding authentication.

Auth-Method6: 1 byte in length, for IPv6 scenario, it indicates the authentication on this interface; this field is defined as a bitmap, this document defines following values (other bits are

reserved and MUST be sent as zero and ignored on receipt):

0x1: PPPoE authentication;  
 0x2: DOT1X authentication;  
 0x4: Web authentication;  
 0x8: Web fast authentication;  
 0x10: Binding authentication;

sub-TLVs:

The IF-Desc sub-TLV can be carried.

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

The flags field is defined as follows:

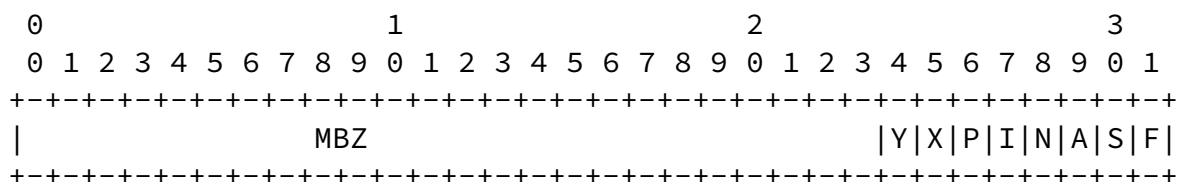


Figure 43: Interface Flags

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. IPoE 1: Enable traffic detection. 0: Disable traffic detection.

P (PPP-Flow-Check) bit: Used for UP detection. PPP 1: Enable traffic detection. 0: Disable traffic detection.

X (ARP-Proxy) bit: 1: The interface is enabled with ARP proxy and can process ARP requests across different Port+VLANs. 0: The ARP proxy is not enabled on the interface, and only the ARP requests of the same Port+VLAN are processed.

Y (ND-Proxy) bit: 1: The interface is enabled with ND proxy and can process ND requests across different Port+VLANs. 0: The ND proxy is not enabled on the interface, and only the ND requests of the same Port+VLAN are processed.

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

## [7.8](#) Routing TLVs

Normally, 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



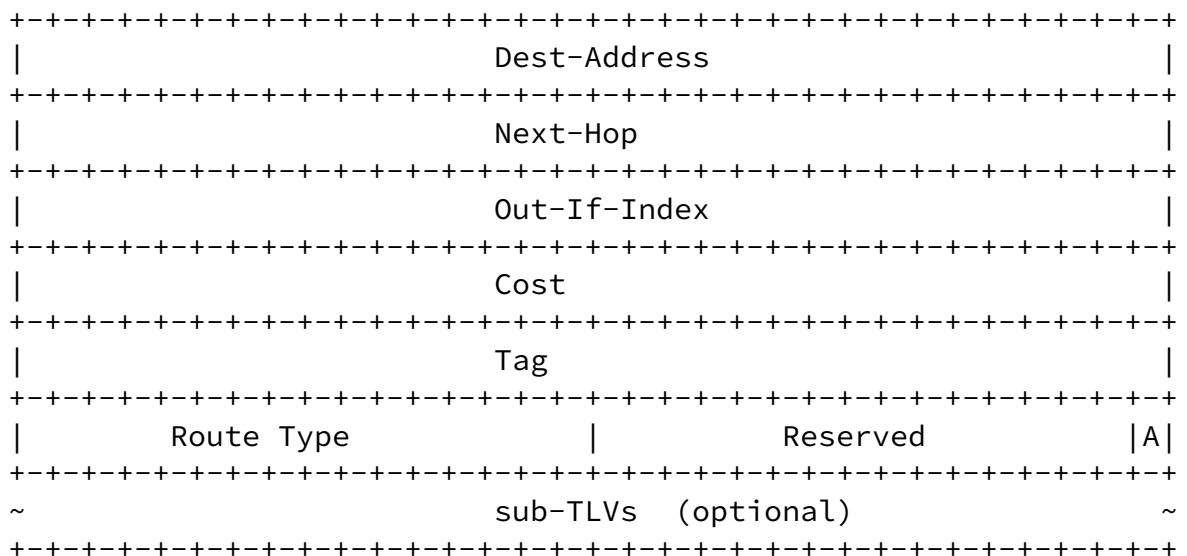


Figure 44: IPv4 Routing TLV

Where:

The TLV Type: 7

The TLV Length: Variable

User-ID: 4 bytes in length. Carries the user identifier. This field 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): Indicates the route type. The options are as follows:



- 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

Advertise-Flag: 1 bit. Indicates whether the route should be advertised by the UP. Following flags 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.

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 this TLV is as follows:

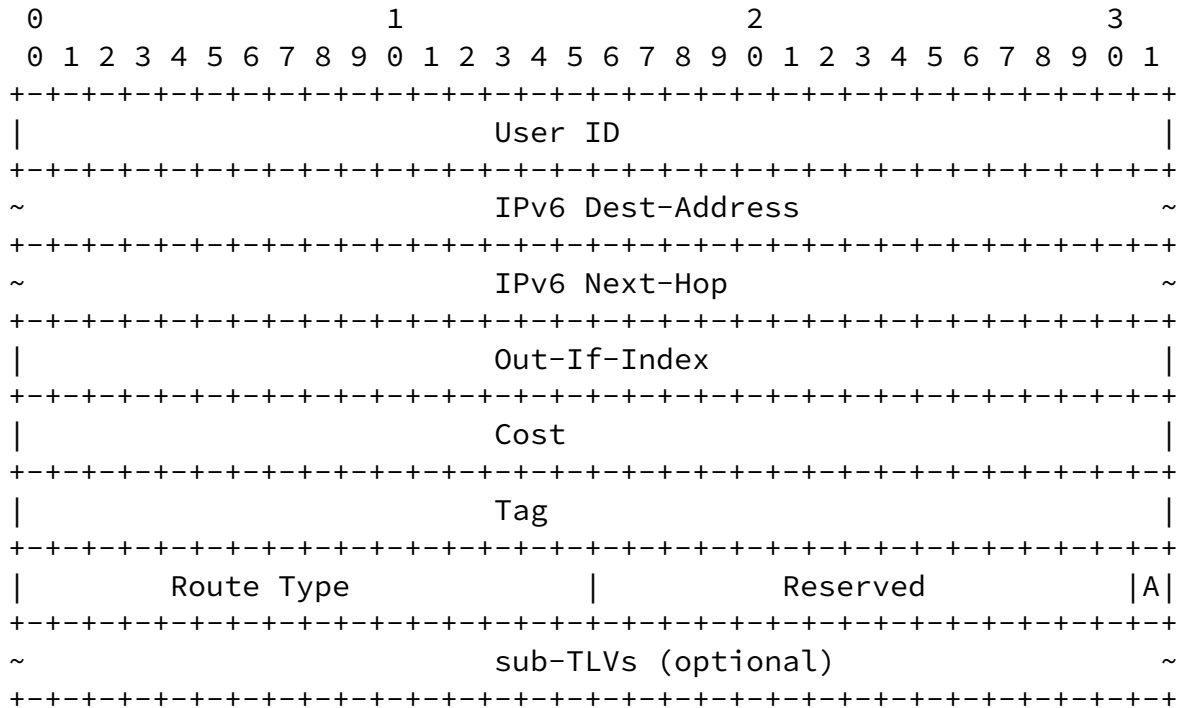


Figure 45: IPv6 Routing TLV

Where:

The TLV Type: 7

The TLV Length: Variable

User-ID: 4 bytes in length. 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): The cost value of the route.

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

Route-Type: (2 bytes): Indicates the route type. The options are as follows:

- 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

Advertise-Flag: 1 bit. Indicates whether the route should be advertised by the UP. Following flags are defined:

- 0: Not advertised,
- 1: advertised.

sub-TLVs: 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.

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 basic 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:

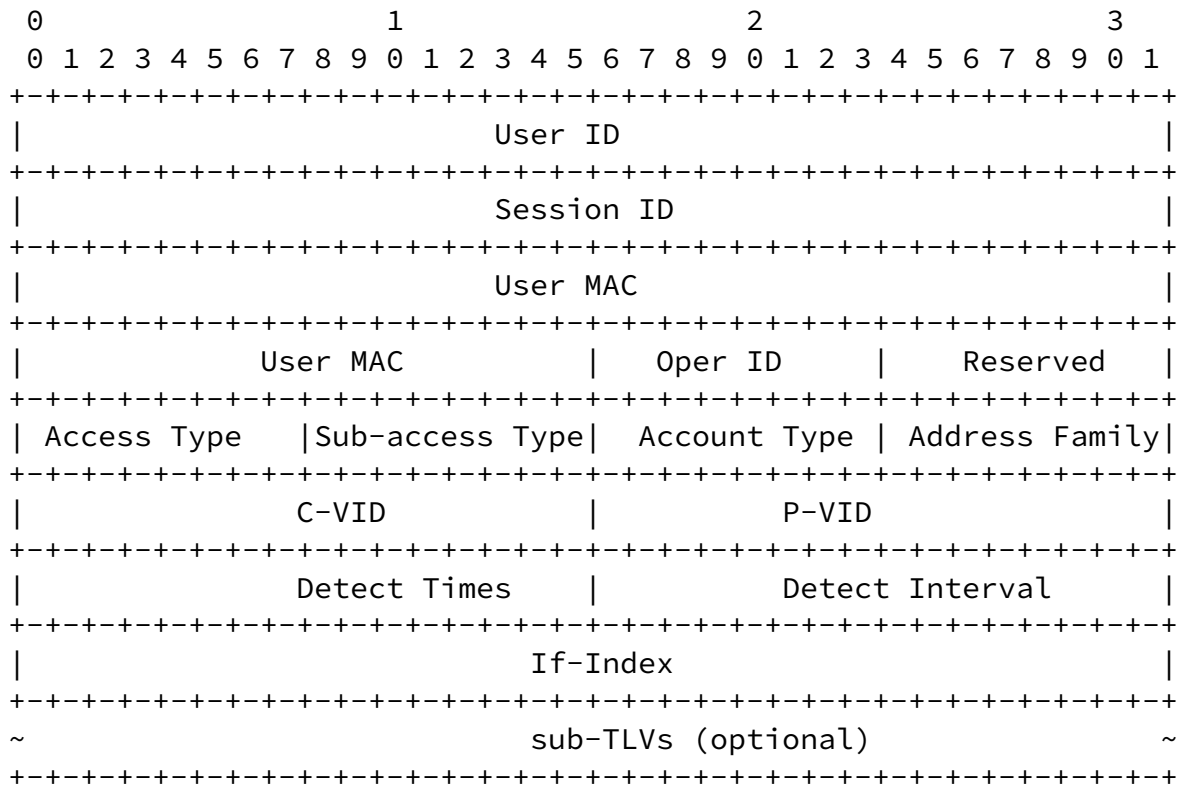


Figure 46: Basic Subscriber TLV

Where:

The TLV Type: 2.

The TLV Length: variable in length.

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

Session-ID (4 bytes): Session ID of a PPPoE subscriber. Zero means 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):

- 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)

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):

- 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)

- 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: 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.

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:

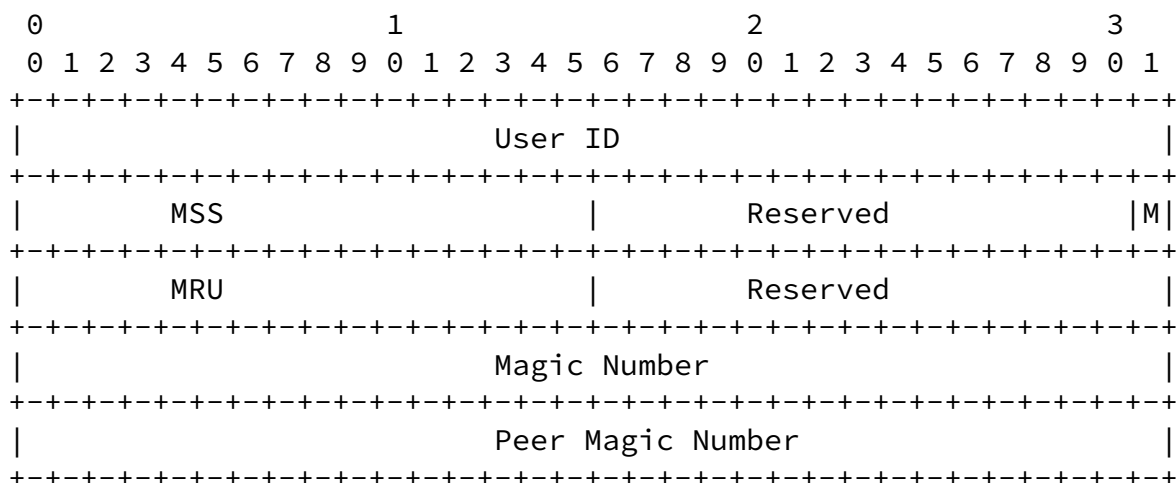


Figure 47: PPP Subscriber TLV

Where:

The TLV type: 3.

The 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.

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:

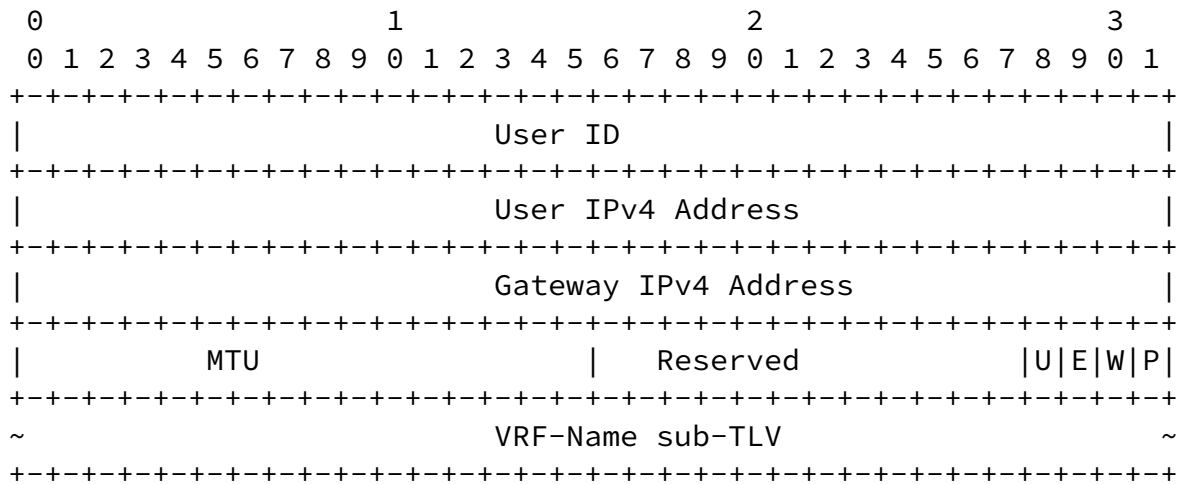


Figure 48: IPv4 Subscriber TLV

Where:

The TLV type: 4.

The 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.

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:

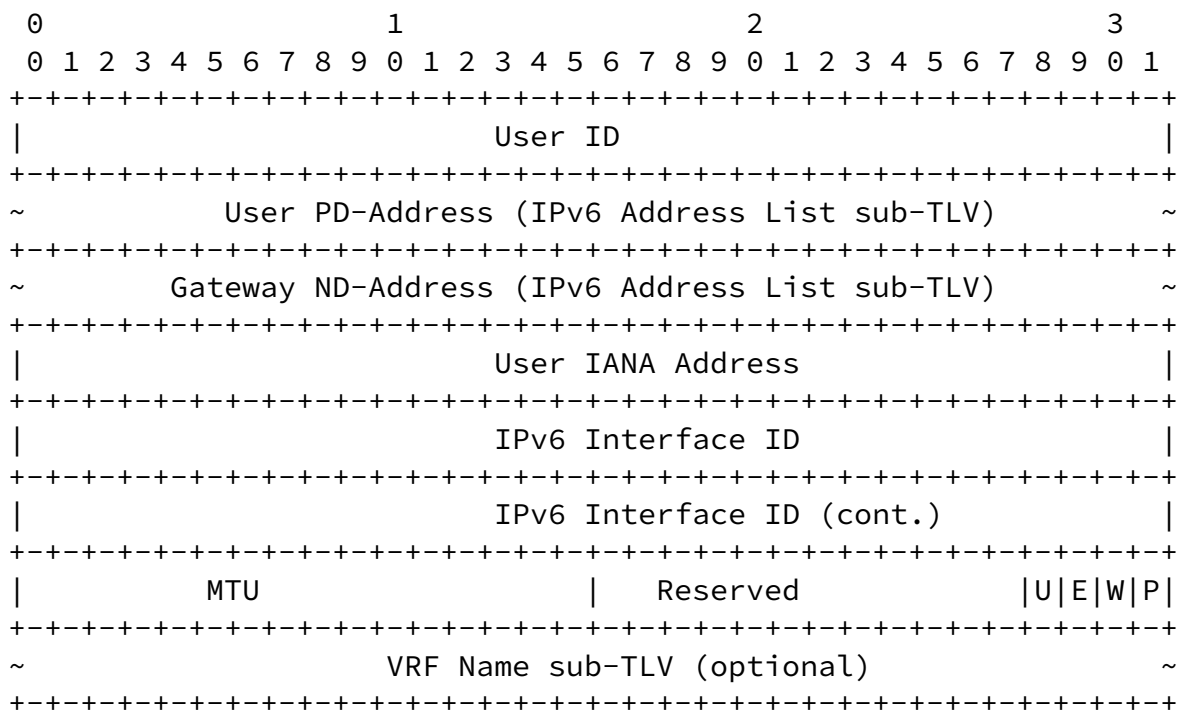


Figure 49: IPv6 Subscriber TLV

Where:

The TLV type: 5.

The TLV length: variable.



INTERNET-DRAFT

Simple BNG CUSP

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 IANA-Address (IPv6-Address): The IANA 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.

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:

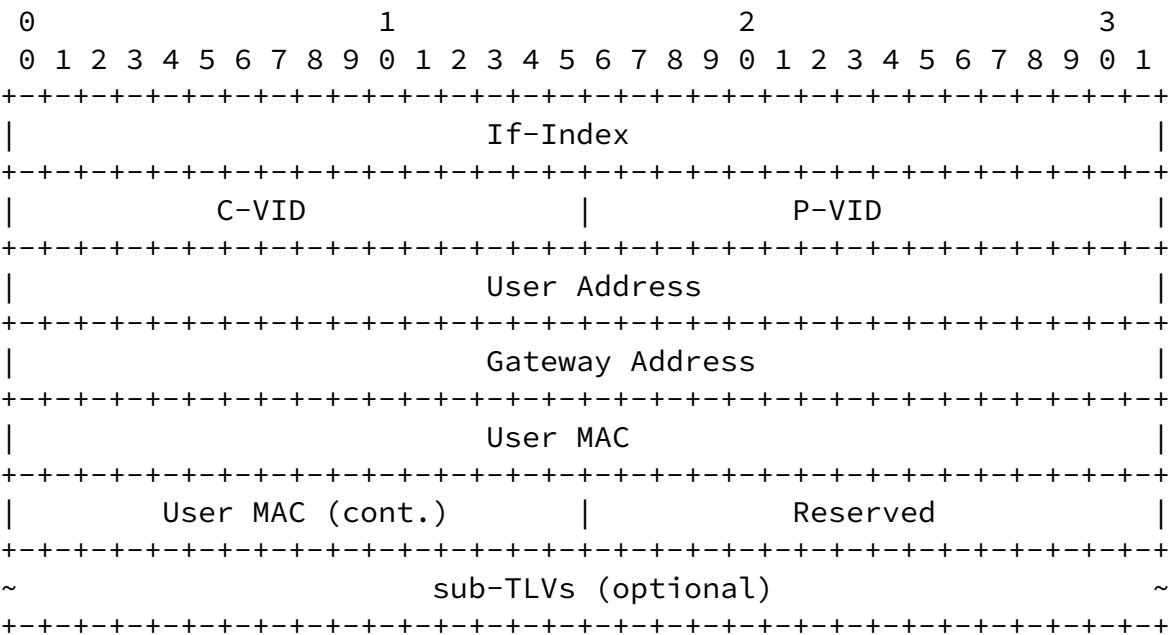


Figure 50: IPv4 Static Subscriber TLV

Where:

The TLV type: 6.

The 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. For a single-layer VLAN, set this parameter to PeVid.

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

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

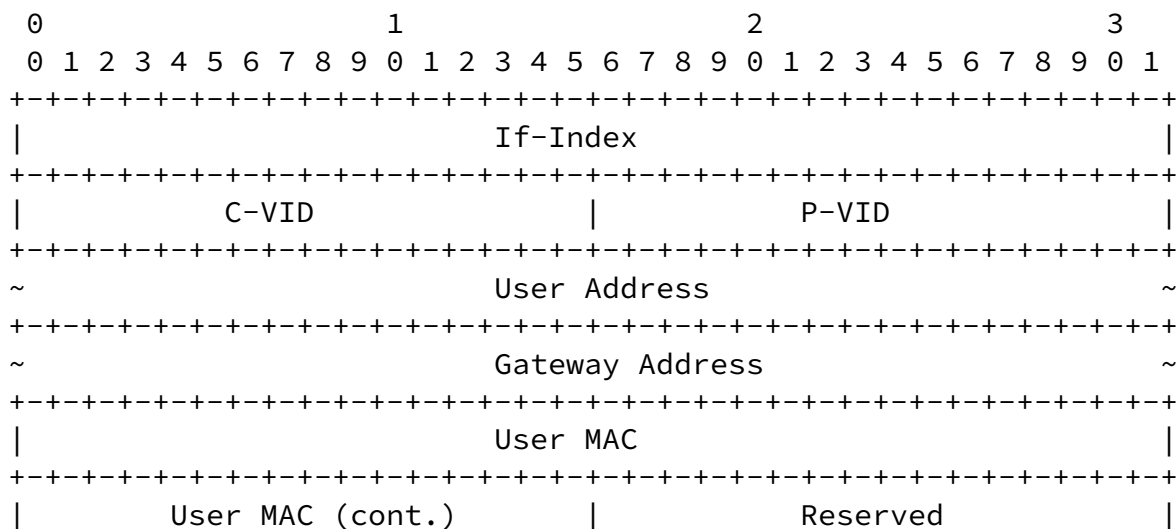
If-Desc sub-TLV: Carries the interface information.

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:



```

+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
~                                         sub-TLVs (optional)                                         ~
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+

```

Figure 51: IPv6 Static Subscriber Detect TLV

Where:

The TLV type: 6.

The 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 C-VID. A valid value is 1~4094. For a single-layer VLAN, set this parameter to PeVid.

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.

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 a 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:

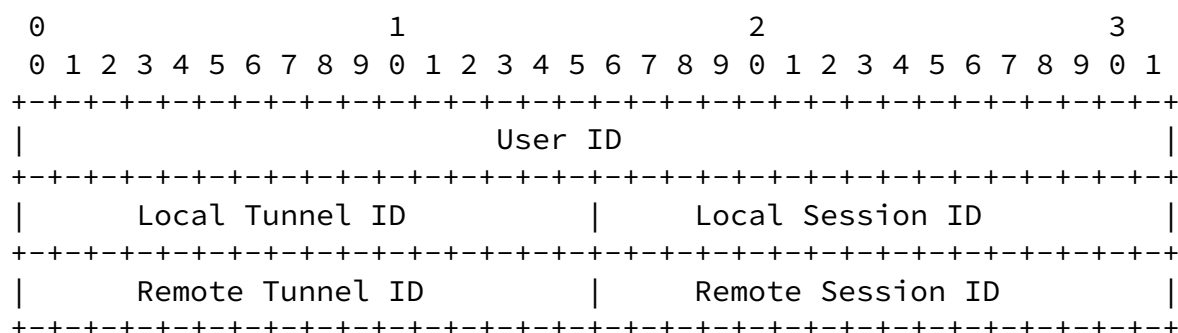


Figure 52: L2TP-LAC Subscriber TLV

Where:

The TLV type: 11.

The 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 remote ID of the L2TP tunnel.

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

#### [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:

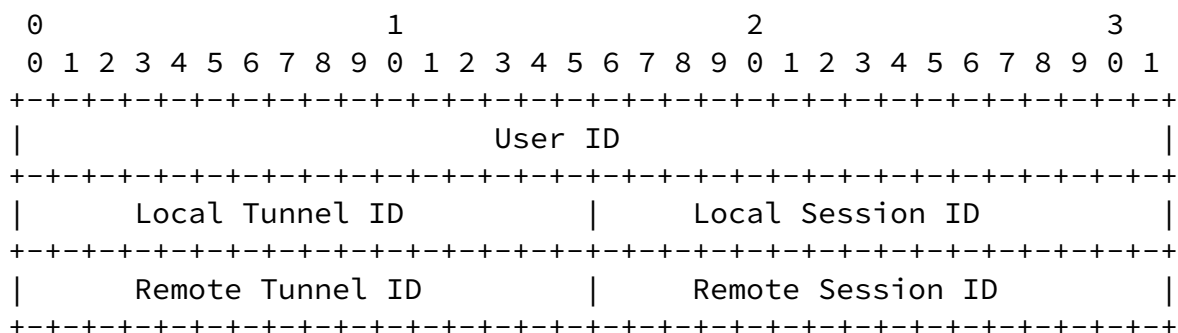


Figure 53: L2TP-LNS Subscriber TLV

Where:

The TLV type: 12.

The 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 remote ID of the L2TP tunnel.

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

#### [7.9.9](#) L2TP-LAC Tunnel TLV

The L2TP-LAC Tunnel TLV is defined to carry the L2TP LAC tunnel related information. 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:



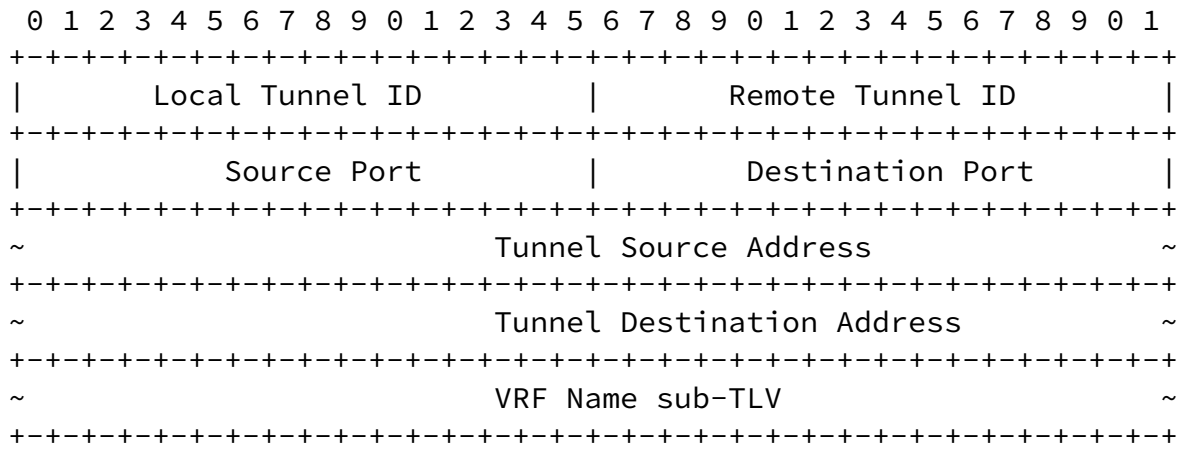


Figure 54: L2TP-LAC Tunnel TLV

Where:

The TLV type: 13.

The 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 the L2TP LNS tunnel related information. 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:

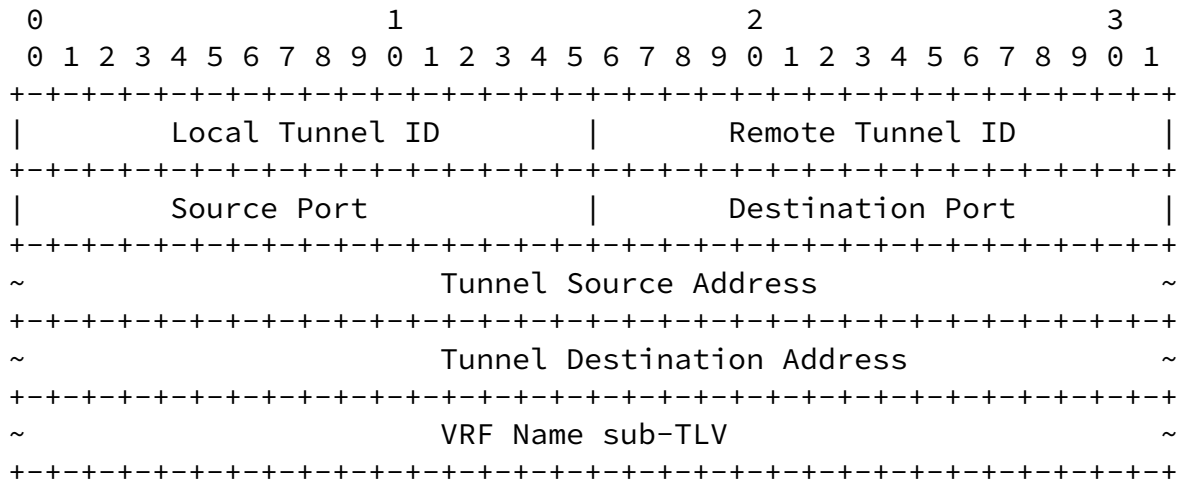


Figure 55: L2TP-LNS Tunnel TLV

Where:

The TLV type: 14.

The 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 Update Response TLV is as follows:

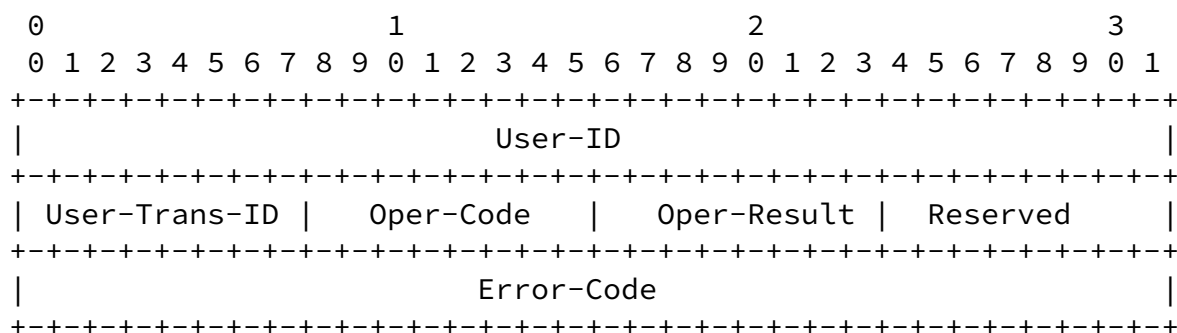


Figure 56: Update Response TLV

Where:

The TLV type: 302.

The TLV length: 12.

User-ID (4 bytes): A unique identifier of an 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 9.5](#).

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:

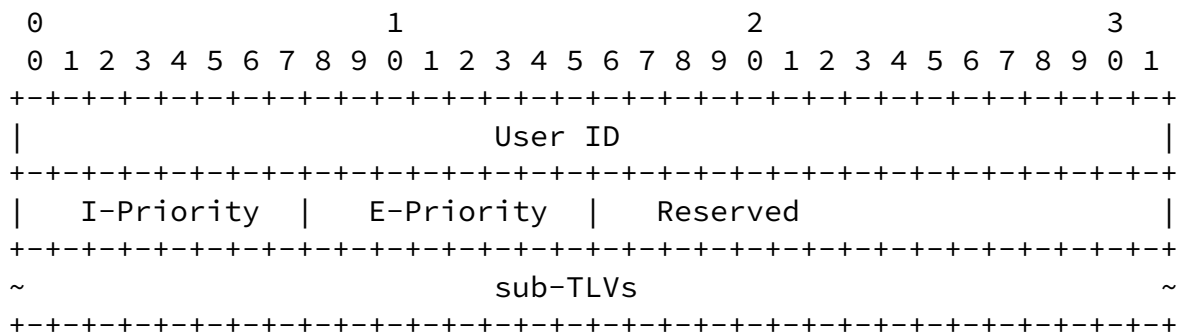


Figure 57: User QoS Policy Information TLV

Where:

The TLV type: 6.

The 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 profile in the upstream direction.

Egress-QoS-Profile Sub-TLV: Indicates the name of the QoS-Profile 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.

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 this TLV is as follows:

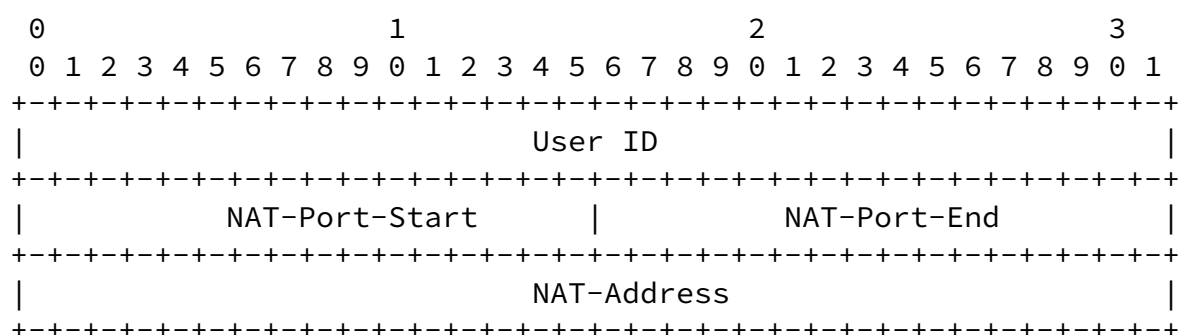


Figure 58: Subscriber CGN Port Range TLV

Where:

The TLV type: 15.

The 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.

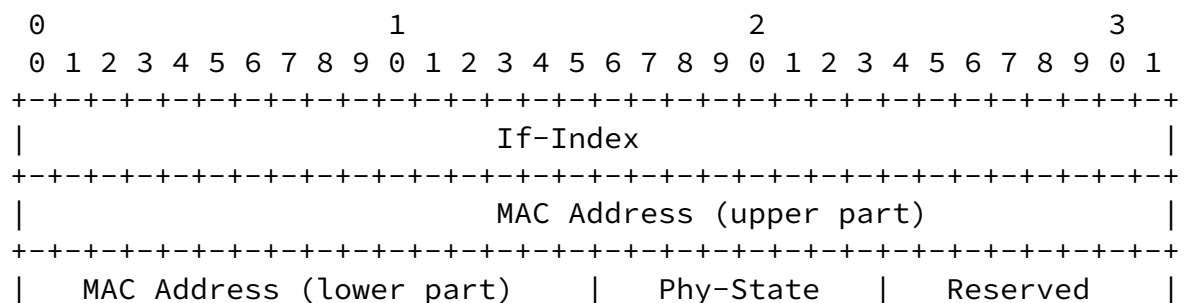
### 7.10 Device Status TLVs

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:







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

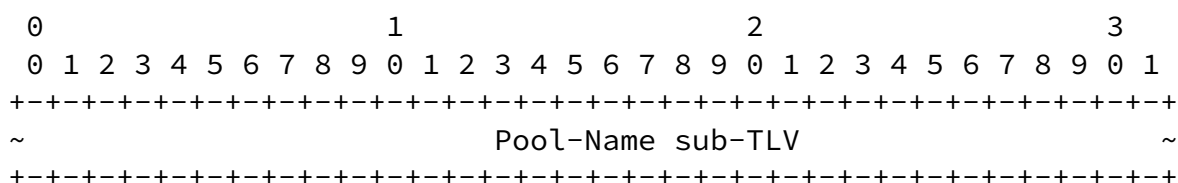


Figure 61: Address Allocation Request TLV

Where:

The TLV type: 400.

The 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 the Addr\_Allocation\_Ack message.

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

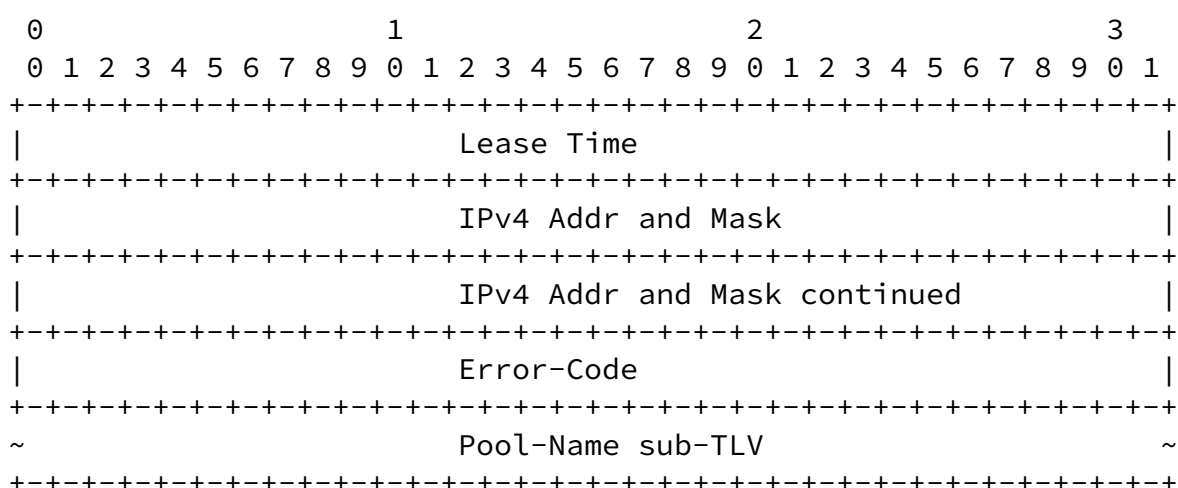


Figure 62: Address Assignment Response TLV

Where:

The TLV type: 401.

The TLV length: variable.

INTERNET-DRAFT

## Simple BNG CUSP

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

IPv4 Addr and Mask (IPv4-Address type): The allocated IPv4 address.

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: A Pool-Name sub-TLV to indicate 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 the Addr\_Renew\_Req message.

The format of this TLV value is as follows:

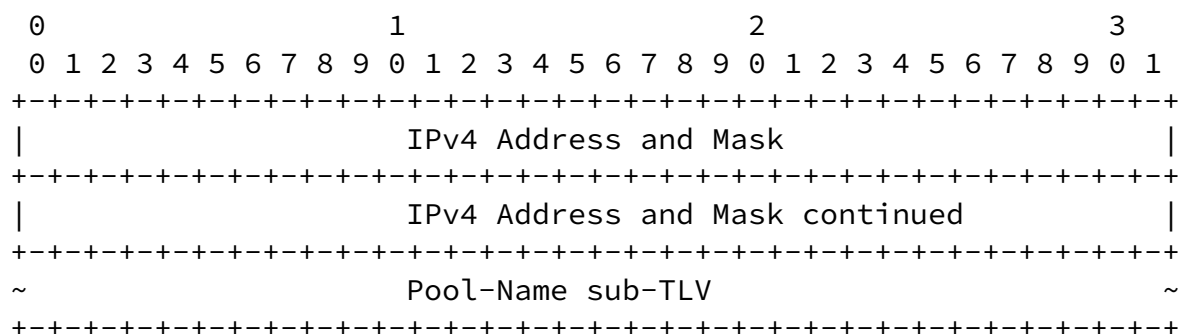


Figure 63: Address Renewal Request TLV

Where:



The TLV type: 402.

The TLV length: variable.

IPv4 Addr and Mask (IPv4-Addr): The IPv4 address to be renewed.

Pool Name sub-TLV: A Pool-Name sub-TLV to indicate from which

Address pool to renew the address.

#### [7.11.4](#) The 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:

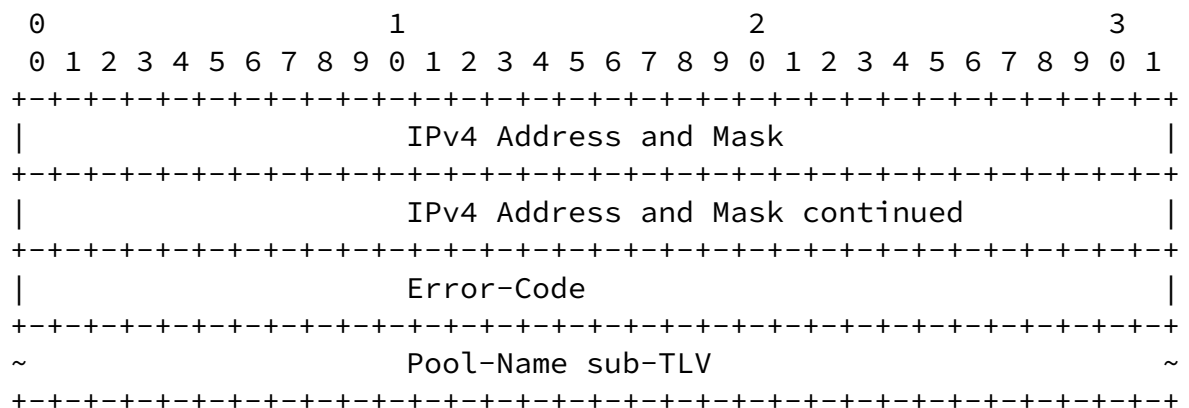


Figure 64: Address Renewal Response TLV

Where:

The TLV type: 403.

The TLV length: variable.

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

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

Where:

The TLV type: 404.

The TLV length: variable.

IPv4 Address and Mask (IPv4-Address type): The IPv4 address be released.

Pool-Name sub-TLV: A Pool-Name Sub-TLV to indicate from which Address pool to release the address.

7.11.6 The 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 this TLV is as follows:

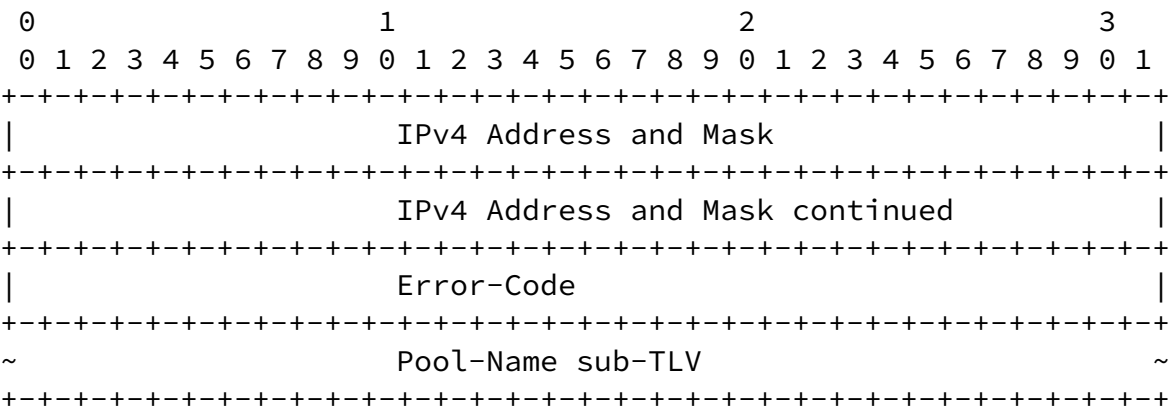


Figure 66: Address Renewal Response TLV

Where:

The diagram shows a horizontal line representing a 40-bit field. Above the line, the segments are labeled 0, 1, 2, and 3. Below the line, the bits are numbered 0 through 39. The line is divided into four equal parts by vertical bars, each representing a 10-bit segment. The label 'User-ID' is centered below the line.

Statistics Type
Ingress Packets (upper part)
Ingress Packets (lower part)
Ingress Bytes (upper part)
Ingress Bytes (lower part)
Ingress Loss Packets (upper part)
Ingress Loss Packets (lower part)
Ingress Loss Bytes (upper part)
Ingress Loss Bytes (lower part)
Egress Packets (upper part)
Egress Packets (lower part)
Egress Bytes (upper part)
Egress Bytes (lower part)
Egress Loss Packets (upper part)
Egress Loss Packets (lower part)
Egress Loss Bytes (upper part)
Egress Loss Bytes (lower part)

Figure 67: Subscriber Traffic Statistics TLV

Where:

The TLV type: 300.

The TLV length: 72 octets.

User-ID (4 bytes): The user/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 upstream direction.

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

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

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

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

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

Egress Loss Packets (8 bytes): The number of the lost packets in 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 a subscriber is online or not. 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).

[illegible]

Figure 68: Subscriber Detection Result TLV

The TLV type: 301.

The TLV length: 8 octets.

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

Detect-Type (1 byte):

- ```
0: IPv4 detection.  
1: IPv6 detection.  
2: PPP detection.
```

Detect-Result (1 bytes):

- 0: Indicates that the detection is successful.
- 1: Detection failure. The UP needs report only when the detection fails.

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

### 7.13 Vendor TLV

The Vendor ID 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:

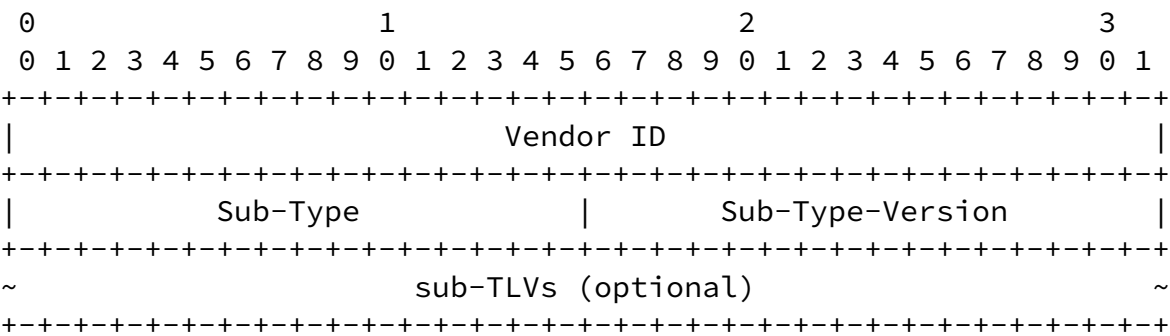


Figure 69: Vendor TLV

Where:

- The TLV type: 1024.
- The TLV length: variable.
- Vendor-ID (4 bytes): Vendor ID ass 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 value 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.](#) Implementation Status

RFC Editor: Please remove this section before publication.

This section discusses the status of implementations that have provided information and the testing of this protocol at the time of posting of this Internet-Draft, and is based on the proposal in [[RFC7942](#)] ("Improving Awareness of Running Code: The Implementation Status Section"). The description of implementations in this section is intended to assist in processing drafts to RFCs.

Please note that the listing of any individual implementation or test results here does not imply endorsement by the RFC Editor or the IETF. Furthermore, no effort has been spent to verify the information presented here that was supplied by contributors. This is not intended as, and must not be construed to be, a catalog of available implementations or their testing or features. Readers are advised to note that other implementations may exist.

According to [[RFC7942](#)], "this will allow reviewers ... to assign due consideration to documents that have the benefit of running code, which may serve as evidence of valuable experimentation and feedback that have made the implemented protocols more mature."

### [8.1](#) Implementations

Information on three S-CUSP implementations appears below.

#### [8.1.1](#) Huawei Technologies

Name: Cloud-based BNG.

Maturity: Production.

Coverage: According to S-CUSP protocol.

Contact information:

Zhouyi Yu: yuzhouyi@huawei.com

Date: 2018-11-01

#### [8.1.2](#) ZTE

Name: ZXR10 V6000 vBRAS

Maturity: Production

Coverage: According to S-CUSP protocol.

Contact information:

Yong Chen: 10056167@zte.com.cn

Huaibin Wang: 10008729@zte.com.cn

Date: 2018-12-01

#### [8.1.3](#) H3C

Name: CUSP protocol for BRAS Control Plane and User Plan Separation

Maturity: Research

Coverage: According to S-CUSP protocol

Contact information: mengdan@h3c.com; liuhanlei@h3c.com

Date: 2019-1-30

## [8.2](#) Hackathon

Successful use of the protocol at the IETF-102 Hackathon, Montreal, Quebec, in 2018.

Hackathon Project: Control Plane and User Plane Separation BNG control channel Protocol (CUSP)

Champions: Zhenqiang Li, Michael Wang

Report: See

[github.com/IETF-Hackathon/ietf102-project-presentations/blob/master/IETF102-hackathon-presentation-CUSP.pptx](https://github.com/IETF-Hackathon/ietf102-project-presentations/blob/master/IETF102-hackathon-presentation-CUSP.pptx)

## [8.3](#) EANTC Testing

EANTC (European Advanced Networking Test Center ([www.eantc.de](http://www.eantc.de))) tested the Huawei implementation. Their summary was as follows: "EANTC tested advanced aspects of the Cloud-based Broadband Network Gateway (vBNG) with a focus on performance, scalability and high availability with up to 20 Million emulated subscribers. The solution performed very well across all test scenarios."

See report at

[www.eantc.de/fileadmin/eantc/downloads/News/2018/EANTC-vBRAS-phase2.pdf](http://www.eantc.de/fileadmin/eantc/downloads/News/2018/EANTC-vBRAS-phase2.pdf)

## [9](#). IANA Considerations

IANA is requested to create an "S-CUSP Parameters" web page and include thereon the registries set up in the Sub-Sections below.

## [9.1](#) Message Types

IANA is requested to create an S-CUSP Message Types registry on the S-CUSP Parameters Web Page as follows:

Registry Name: S-CUSP Message Types  
Registration Procedure: Expert Review  
Reference: [this document]

| 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          |                            |

## [9.2](#) TLV Types

IANA is requested to create an S-CUSP TLV Types registry on the S-CUSP Parameters Web Page as follows:

Registry Name: S-CUSP TLV Types  
 Registration Procedure: Expert Review  
 Reference: [this document]

| 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 Keep Alive timers negotiation.                                          |
| 101   | Error Information             | Carried in the Ack of the control message. Carries the processing result, success, or error. |
| 102   | Keep Alive                    | Carried in the Hello message for Keep Alive timers                                           |

INTERNET-DRAFT

Simple BNG CUSP

|           |                               |                                                                                                                              |
|-----------|-------------------------------|------------------------------------------------------------------------------------------------------------------------------|
| 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 Results  | User detection information.                                                                                                  |
| 302       | Update Response               | The processing result of a subscriber session update.                                                                        |
| 303-299   | unassigned                    | -                                                                                                                            |
| 400       | Address Allocation Request    | Request address allocation.                                                                                                  |
| 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 vendor.                                                                                                    |
| 1039-4095 | unassigned                    | -                                                                                                                            |

### [9.3](#) TLV Operation Codes

IANA is requested to create an S-CUSP TLV Operation Codes registry on the S-CUSP Parameters Web Page as follows:

Registry Name: S-CUSP TLV Operation Codes  
 Registration Procedure: Expert Review  
 Reference: [this document]

| Code | Operation |
|------|-----------|
|------|-----------|

```

-----
0    - reserved
1    Update
2    Delete
3-15 - unassigned

```

#### [9.4](#) Sub-TLV Types

IANA is requested to create an S-CUSP Sub-TLV Types registry on the S-CUSP Parameters Web Page as follows:

Registry Name: S-CUSP Sub-TLV Types  
 Registration Procedure: Expert Review  
 Reference: [this document]

| 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          |                            |

#### [9.5](#) Error Codes

IANA is requested to create an S-CUSP ERRID Codes registry on the S-



CUSP Parameters Web Page as follows:

Registry Name: S-CUSP ERRID Codes  
Registration Procedure: Expert Review  
Reference: [this document]

| Value | Name             | Remarks                                     |
|-------|------------------|---------------------------------------------|
| ----- | -----            | -----                                       |
| 0     | Success          | Success                                     |
| 1     | Fail             | 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.   |

Hu, et al

[Page 118]

---

INTERNET-DRAFT

Simple BNG CUSP

|           |                    |                                                                             |
|-----------|--------------------|-----------------------------------------------------------------------------|
|           |                    | 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 forwarding table delivery error codes.

5000-4294967295 - reserved

## [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 a network operator decision. See [[RFC6241](#)] for Management Interface / NETCONF security. Security on the Service Interface is dependent on the tunneling protocol used which is out of scope for this document. Security for the Control Interface, over which the S-CUSP protocol 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 Control Interface 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.

#### Contributors

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

- [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>>.

INTERNET-DRAFT

Simple BNG CUSP

## Informative References

- [802.1Q] IEEE, "IEEE Standard for Local and metropolitan area networks / Bridges and Bridged Networks", IEEE Std 802.1Q-2014, 3 November 2013.
- [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>>.

[RFC7942] Sheffer, Y. and A. Farrel, "Improving Awareness of Running Code: The Implementation Status Section", [BCP 205](#), [RFC 7942](#), DOI 10.17487/RFC7942, July 2016, <<https://www.rfc-editor.org/info/rfc7942>>.

[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, 2018.

---

INTERNET-DRAFT

Simple BNG CUSP

Authors' Addresses

Shujun Hu  
China Mobile  
32 Xuanwumen West Ave, Xicheng District  
Beijing, Beijing 100053  
China

Email: [hushujun@chinamobile.com](mailto:hushujun@chinamobile.com)

Donald Eastlake, 3rd  
Futurewei Technologies  
1424 Pro Shop Court  
Davenport, FL 33896  
USA

Phone: +1-508-333-2270  
Email: [d3e3e3@gmail.com](mailto:d3e3e3@gmail.com)

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

Email: [mach.chen@huawei.com](mailto:mach.chen@huawei.com)



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

Email: qinfengwei@chinamobile.com

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

Email: lizhenqiang@chinamobile.com

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

Email: teemong@singtel.com

Daniel Huang  
ZTE

Email: huang.guangping@zte.com.cn

---

INTERNET-DRAFT

Simple BNG CUSP

#### Copyright, Disclaimer, and Additional IPR Provisions

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](http://trustee.ietf.org/license-info) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must

include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.