The China Mobile, Huawei, and ZTE BNG
Simple Control and User Plane Separation Protocol (S-CUSP)
draft-chz-simple-cu-separation-bng-protocol-06

Abstract

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

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

Status of This Memo

This 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.

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."
The list of current Internet-Drafts can be accessed at http://www.ietf.org/1id-abstracts.html. The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.
# Table of Contents

1. Introduction........................................... 6
2. Terminology............................................ 7
   2.1. Implementation Requirement Keywords.............. 7
   2.2. Terms............................................. 7
3. BNG CUPS Overview.................................... 10
   3.1. BNG CUPS Motivation............................. 10
   3.2. BNG CUPS Architecture Overview................ 10
   3.3. BNG CUPS Interfaces............................. 12
   3.3.1. Service Interface............................... 13
   3.3.2. Control Interface............................... 14
   3.3.3. Management Interface........................... 14
   3.4. BNG CUPS Procedure Overview.................... 14
4. S-CUSP Protocol Overview.............................. 18
   4.1. Control Channel Related Procedures............... 18
   4.1.1. S-CUSP Session Establishment.................. 18
   4.1.2. Keepalive Timer and DeadTimer................ 19
   4.2. Node Related Procedures.......................... 20
   4.2.1. UP Resource Report............................. 20
   4.2.2. Update BAS Function on Access Interface......... 21
   4.2.3. Update Network Routing........................ 21
   4.2.4. CGN Public IP Address Allocation.............. 22
   4.2.5. Data Synchronization between the CP and UP...... 23
   4.3. Subscriber Session Related Procedures............ 24
   4.3.1. Create Subscriber Session..................... 25
   4.3.2. Update Subscriber Session..................... 26
   4.3.3. Delete Subscriber Session..................... 27
   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............................................. 48
   5.4.1. L2TP LAC Access................................ 48
   5.4.2. L2TP LNS IPv4 Access............................ 49
   5.4.3. L2TP LNS IPv6 Access............................ 51
   5.5. CGN (Carrier Grade NAT).......................... 54
   5.6. L3 Leased Line Access............................. 55
   5.6.1. Web Authentication.............................. 55
   5.6.2. User Traffic Trigger............................ 57
   5.7. Multicast Service Access........................... 59
Table of Contents (continued)

6.  S-CUSP Message Formats........................................61
   6.1.  Common Message Header....................................61
   6.2.  Control Messages...........................................62
       6.2.1.  Hello Message........................................62
       6.2.2.  Keepalive Message....................................63
       6.2.3.  Sync_Request Message.................................63
       6.2.4.  Sync_Begin Message..................................64
       6.2.5.  Sync_Data Message....................................64
       6.2.6.  Sync_End Message.....................................65
       6.2.7.  Update_Request Message...............................65
       6.2.8.  Update_Response Message..............................66
   6.3.  Event Message..............................................67
   6.4.  Report Message..............................................67
   6.5.  CGN Messages................................................67
       6.5.1.  Addr_Allocation_Req Message.........................67
       6.5.2.  Addr_Allocation_Ack Message........................68
       6.5.3.  Addr_Renew_Req Message..............................68
       6.5.4.  Addr_Renew_Ack Message..............................68
       6.5.5.  Addr_Release_Req Message............................68
       6.5.6.  Addr_Release_Ack Message............................68
   6.6.  Vendor Message..............................................69
   6.7.  Error Message..............................................69

7.  S-CUSP TLVs and Sub-TLVs......................................70
   7.1.  Common TLV Header...........................................70
   7.2.  Basic Data Fields..........................................71
   7.3.  Sub-TLV Format and Sub-TLVs................................72
       7.3.1.  Name sub-TLVs........................................72
       7.3.2.  Ingress-CAR sub-TLV................................73
       7.3.3.  Egress-CAR sub-TLV..................................73
       7.3.4.  If-Desc sub-TLV......................................74
       7.3.5.  IPv6 Address List sub-TLV............................76
       7.3.6.  Vendor sub-TLV........................................76
       7.4.  The Hello TLV.............................................77
       7.5.  The Keepalive TLV........................................79
       7.6.  The Error Information TLV...............................80
       7.7.  BAS Function TLV.........................................80
       7.8.  Routing TLVs...............................................82
           7.8.1.  IPv4 Routing TLV................................83
           7.8.2.  IPv6 Routing TLV................................84
       7.9.  Subscriber TLVs..........................................86
           7.9.1.  Basic Subscriber TLV...............................86
           7.9.2.  PPP Subscriber TLV................................88
           7.9.3.  IPv4 Subscriber TLV................................89
           7.9.4.  IPv6 Subscriber TLV................................91
           7.9.5.  IPv4 Static Subscriber Detect TLV...............92
           7.9.6.  IPv6 Static Subscriber Detect TLV...............93
           7.9.7.  L2TP-LAC Subscriber TLV..........................95
           7.9.8.  L2TP-LNS Subscriber TLV..........................95
           7.9.9.  L2TP-LAC Tunnel TLV...............................96
Table of Contents (continued)

7.9.10. L2TP-LNS Tunnel TLV.................................97
7.9.11. Update Response TLV.................................98
7.9.12. Subscriber Policy TLV...............................99
7.9.13. Subscriber CGN Port Range TLV.....................101
7.10. Device Status TLVs....................................101
7.10.1. Interface Status TLV................................102
7.10.2. Board Status TLV...................................102
7.11. CGN TLVs.............................................103
7.11.1. Address Allocation Request TLV......................103
7.11.2. Address Allocation Response TLV....................104
7.11.3. Address Renewal Request TLV.........................105
7.11.4. The Address Renewal Response TLV..................106
7.11.5. Address Release Request TLV........................107
7.11.6. The Address Release Response TLV..................107
7.12. Event TLVs..........................................108
7.12.2. Subscriber Detection Result TLV....................110
7.13. Vendor TLV..........................................111
8. Implementation Status......................................113
8.1. Implementations........................................113
8.1.1. Huawei Technologies................................113
8.1.2. ZTE...............................................114
8.1.3. H3C..............................................114
8.2. Hackathon.............................................114
8.3. EANTC Testing.........................................115
9. Tables of S-CUSP Codepoints..............................116
9.1. Message Types........................................116
9.2. TLV Types............................................116
9.3. TLV Operation Codes...................................118
9.4. Sub-TLV Types........................................119
9.5. Error Codes..........................................119
9.6. If-Type Values.......................................120
9.7. Access-Mode Values...................................121
9.8. Access Method Bits....................................121
9.9. Route-Type Values....................................122
9.10. Access-Type Values..................................122
10. IANA Considerations......................................123
11. Security Considerations.................................124

Contributors................................................125
Acknowledgements...........................................127
Normative References.......................................128
Informative References.....................................129
Authors’ Addresses.........................................131
1. Introduction

A Broadband Network Gateway (BNG) in a fixed wireline access network is an Ethernet-centric IP edge router, and the aggregation point for subscriber traffic. To provide centralized session management, flexible address allocation, high scalability for subscriber management capacity, and cost-efficient redundancy, the Control/User (CU) separated BNG framework is described in technical report [TR-384] from the Broadband Forum (BBF). 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.

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

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 allow for easy addition of messages and data items, should further requirements be expressed in the future.

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

At the time of writing this document, the Broadband Forum (BBF) is working to produce [WT-459] that will describe an architecture and requirements for a control and user plane separation of a disaggregated BNG. Future work may attempt to show how the protocol described in this document addresses those requirements and may modify this specification to handle unaddressed requirements.
2. Terminology

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

2.1. Implementation Requirement Keywords

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

2.2. Terms

This section specifies terms used in this document.


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

EMS: Element Management System.

IPoE: IP over Ethernet.

L2TP: Layer 2 Tunneling Protocol [RFC2661].

LAC: L2TP Access Concentrator.

LNS: L2TP Network Server.

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

MANO: Management and Orchestration.

Mi: Management Interface.

MSS: Maximum Segment Size.

MRU: Maximum Receive Unit.

NAT: Network Address Translation [RFC3022].

ND: Neighbor Discovery.

NFV: Network Function Virtualization.
NFVI: NFV Infrastructure

PBS: Peak Burst Size.

PD: Prefix Delegation.

PIR: Peak Information Rate.

PPP: Point to Point Protocol [RFC1661].

PPPoE: PPP over Ethernet [RFC2516].

RBNF: Routing Backus-Naur Form [RFC5511].

RG: Residential Gateway.


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

Si: Service Interface.

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

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

URPF: Unicast Reverse Path Forwarding.

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

VRF: Virtual Routing and Forwarding.
3. BNG CUPS Overview

3.1. BNG CUPS Motivation

The rapid development of new services, such as 4K TV, 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.

The framework for a cloud-based BNG with Control Plane and User Plane (CU) separation to address these challenges for fixed networks is described in [TR-384]. The main idea of CU separation is to extract and centralize the user management functions of multiple BNG devices, forming a unified and centralized 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 deployed as a centralized 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:

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 external 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

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

Hu, et al
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 that can be configured 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.

<table>
<thead>
<tr>
<th>RG</th>
<th>UP</th>
<th>CP</th>
<th>AAA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Establish S-CUSP Channel</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>&lt;----------------------&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Report Board</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>interface</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>information</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>------to CP via Ci-----&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Update BAS function</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>request / response</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>&lt;=-----on UP via Ci-----&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Update network routing</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>request / response</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>&lt;=-------- via Ci--------&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Online Req</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.1</td>
<td>----------------------&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Relay the Online Req</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.2</td>
<td>------to CP via Si-----&gt;</td>
<td>Authentication</td>
<td></td>
</tr>
</tbody>
</table>
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-facing interfaces of this UP to the CP. A board can also be called a Line/Service.
Process Unit (LPU/SPU) card. The subscriber-facing interfaces refer to the interfaces that connect the Access Network nodes (e.g., 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 (as defined in Section 2.1) 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.

8.2. 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](image)

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 an S-CUSP session over the TCP connection.

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

The version information (Hello TLV, see Section 7.4) is carried within Hello messages (see Section 6.2.1). A CP can support multiple versions, but a UP can only support one version. So, the version negotiation is based on whether a version can be supported 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.

<table>
<thead>
<tr>
<th>Timer Name</th>
<th>Range in seconds</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Establishment Timer</td>
<td>1-32767</td>
<td>45</td>
</tr>
<tr>
<td>Keepalive Timer</td>
<td>0-255</td>
<td>30</td>
</tr>
<tr>
<td>DeadTimer</td>
<td>1-32767</td>
<td>4 * Keepalive</td>
</tr>
</tbody>
</table>

4.1.2. Keepalive Timer and DeadTimer

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

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

Each end of a S-CUSP session also run a DeadTimer, and restarts that DeadTimer whenever a message is received on the session. If the DeadTimer at an end of the session expires, that end declares the session dead and the session will be closed, unless their DeadTimer is set to the special value zero in which case the session will not time out.
The minimum value of the Keepalive timer is 1 second, and it is specified in units of 1 second. The RECOMMENDED default value is 30 seconds. The recommended default for the DeadTimer is four times the value of the Keepalive timer used by the remote peer. As above, the timers may be disabled by setting them to zero.

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

4.2. Node 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 state information of the Boards and access-facing interfaces on the UP to the CP, as shown in Figure 5. Report messages are unacknowledged and are assumed to be delivered because the session runs over TCP.

The CP can use that information to activate/enable the 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 a 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](image)

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](image)

4.2.3. Update Network Routing

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

![Figure 7: Update Network Routing](image)

The Update Request and Response Message exchanges, carrying the IPv4/IPv6 Routing Information TLVs (Section 7.8), update the subscriber’s network routing information.
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.
### 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

2) Synchronization from CP to UP

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 collection of hardware resources (e.g., NP, TCAM etc.) have to be allocated to a session on a UP as part of its initiation.
Subscriber session related procedures include subscriber session create, update, delete, and statistics report. The following subsections 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.)

<table>
<thead>
<tr>
<th>RG</th>
<th>UP</th>
<th>CP</th>
<th>AAA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Online Request</td>
<td>Relay the Online Request</td>
<td>Authentication</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>to CP via Si</td>
<td>Auth/Req/Rep</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Create subscriber</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>session Request</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Create subscriber</td>
<td></td>
</tr>
<tr>
<td></td>
<td>session Response</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Send Online Response</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Online Response</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Accounting</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**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). A tunneling technology implements the Service Interface.

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

That completes the subscriber activation process.

4.3.2. Update Subscriber Session

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

<table>
<thead>
<tr>
<th>UP</th>
<th>CP</th>
<th>AAA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>COA Request</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>&lt;-------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Session update Request</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>&lt;--------via Ci--------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Session update Response</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>--------via Ci--------&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>COA Response</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>&lt;-------------</td>
</tr>
</tbody>
</table>

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 how S-CUPS deals with a subscriber offline request.

\[
\begin{array}{c|c|c}
\text{RG} & \text{UP} & \text{CP} \\
\hline
\text{Offline Request} & \text{Relay the Offline Request} & \text{Send the Offline Response} \\
1 \xrightarrow{} & 2 \xrightarrow{} \text{to CP via Si} & 3 \xleftarrow{} \text{to UP via Si} \\
\hline
\text{Offline Response} & \text{Session delete Request} & \text{Session delete Response} \\
4 \xleftarrow{} & \xleftarrow{} \text{via Ci} & \xleftarrow{} \text{via Ci} \\
\end{array}
\]

Figure 12: Subscriber Session Delete

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

When the CP receives the offline request, it will withdraw/release the resources (e.g., IP address, bandwidth) that have been allocated to the subscriber. 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

\[
\begin{array}{c|c}
\text{UP} & \text{CP} \\
\hline
\text{Statistic/Detect report} & \xrightarrow{} \text{via Ci} \\
\end{array}
\]

Figure 13: Events Report
When a session is created on a 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 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.

<table>
<thead>
<tr>
<th>RG</th>
<th>UP</th>
<th>CP</th>
<th>AAA</th>
</tr>
</thead>
<tbody>
<tr>
<td>DHCP Discovery</td>
<td>Relay the DHCP Discovery</td>
<td>AAA</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>--------------------</td>
<td>---------------------</td>
<td>----------------------</td>
</tr>
<tr>
<td></td>
<td>2&lt;----to CP via Si---</td>
<td>Req/Rep</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>5&lt;---------------</td>
<td>---------------------</td>
<td></td>
</tr>
<tr>
<td>DHCP Request</td>
<td>Send the DHCP Offer</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>6&lt;---------------</td>
<td>---------------------</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Relay the DHCP Request</td>
<td></td>
</tr>
<tr>
<td></td>
<td>7&lt;----to CP via Si---</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Create subscriber</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>session Request</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>8&lt;--------via Ci------</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Create subscriber</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>session Response</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>9&lt;--------via Ci------</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Accounting</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>10&lt;-----------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Send DHCP ACK</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>11&lt;----to UP via Si---</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The S-CUSP protocol implements steps 8 and 9.

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

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

\[
\text{<Update\_Request\ Message> ::= <Common Header>}
\text{<Basic Subscriber TLV>}
\text{<IPv4 Subscriber TLV>}
\text{<IPv4 Routing TLV>}
\text{[[<Subscriber Policy TLV>]]}
\]

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

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

### 5.1.2. DHCPv6 Access

The following sequence shows detailed procedures for DHCPv6 access.

\[
\begin{array}{l|l|l|l}
\text{RG} & \text{UP} & \text{CP} & \text{AAA} \\
\hline
1 & \text{Request} & \text{Relay the Solicit} & \text{Relay the Solicit} \\
2 & \text{Advertise} & \text{Send the Advertise} & \text{Send the Advertise} \\
4 & \text{Request} & \text{Advertise} & \text{Advertise} \\
\hline
\end{array}
\]
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:

\[
<\text{Update}\_\text{Request Message}> ::= <\text{Common Header}> \quad <\text{Basic Subscriber TLV}> \quad <\text{IPv6 Subscriber TLV}> \quad <\text{IPv6 Routing TLV}> \quad [<\text{Subscriber Policy TLV}>]
\]

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

\[
<\text{Update}\_\text{Response Message}> ::= <\text{Common Header}> \quad <\text{Update Response TLV}>
\]
### 5.1.3. IPv6 SLAAC Access

The following flow shows the IPv6 SLAAC access process.

<table>
<thead>
<tr>
<th>RG</th>
<th>UP</th>
<th>CP</th>
<th>AAA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>RS</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td>RS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Relay the Router Solicit (RS)</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td>AAA Req/Rep</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Create subscriber session Request</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3&lt;--------------------------&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>AAA Req/Rep</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Create subscriber session Response</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5&lt;--------------------------&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>AAA Req/Rep</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Send Router Advertise (RA)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>6&lt;--------------------------&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>AAA Req/Rep</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Relay the Neighbor Solicit (NS)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>9&lt;--------------------------&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>AAA Req/Rep</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Send a Neighbor Advertise (NA)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>11&lt;--------------------------&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>AAA Req/Rep</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Accounting</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>10&lt;--------------------------&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>AAA Req/Rep</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>NA</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>12&lt;--------------------------&gt;</td>
</tr>
</tbody>
</table>

**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.

<table>
<thead>
<tr>
<th>RG</th>
<th>UP</th>
<th>CP</th>
<th>AAA</th>
</tr>
</thead>
<tbody>
<tr>
<td>RS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>RS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

RS 1 ---------->
Relay the Router
Solicit (RA)

2 -----to CP via Si----->
AAA
Req/Rep

3 <---------->
Create subscriber
session Request

4 <----------via Ci---------->
Create subscriber
session Response

5 <----------via Ci---------->
Send Router Advertise
(RA)

6 <----to UP vis Si-------
RA

7 <----------
DHCPv6 Solicit

8 ---------->
Relay DHCPv6 Solicit
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).

\[\text{Update Request Message} ::= \text{<Common Header>}\]
\[\quad \text{<Basic Subscriber TLV>}\]
\[\quad \text{<IPv6 Subscriber TLV>}\]
\[\quad \text{<IPv6 Routing TLV>}\]
\[\quad [\text{<Subscriber Policy TLV>}]\]

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

\[\text{Update Response Message} ::= \text{<Common Header>}\]
\[\quad \text{<Update Response TLV>}\]

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:

\[\text{Update Request Message} ::= \text{<Common Header>}\]
\[\quad \text{<Basic Subscriber TLV>}\]
\[\quad \text{<IPv6 Subscriber TLV>}\]
\[\quad \text{<IPv6 Routing TLV>}\]
\[\quad [\text{<Subscriber Policy TLV>}]\]
The UP will reply with an Update_Response message (step 11). The format of the Update_Response is as follows:

\[
<\text{Update\_Response\ Message}> ::= <\text{Common\ Header}> \\
<\text{Update\ Response\ TLV}>
\]

5.1.5. DHCP Dual Stack Access

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

<table>
<thead>
<tr>
<th>RG</th>
<th>UP</th>
<th>CP</th>
<th>AAA</th>
</tr>
</thead>
<tbody>
<tr>
<td>DHCP Discovery</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>---------------→</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2 ←→ CP via Si ----→</td>
<td></td>
<td>AAA</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td></td>
<td>Req/Resp</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3 ←→ to CP via Si------&gt;</td>
<td>AAA</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>4 ←→ to UP via Si------&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DHCP Request</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>←→----------------------&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DHCP Offer</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>←→----------------------&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>7</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>7 ←→ CP via Si------&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>8 ←→ via Ci --------&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Create subscriber</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>session Request</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>←→----------------------&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>9 ←→ via Ci --------&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Create subscriber</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>session Response</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>←→----------------------&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Accounting</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>←→----------------------&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DHCP ACK</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>←→----------------------&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>←→----------------------&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>14</td>
<td></td>
<td>AAA</td>
</tr>
<tr>
<td>Relay the Router</td>
<td></td>
<td></td>
<td>Req/Resp</td>
</tr>
<tr>
<td>Solicit (RA)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>←→ CP via Si ------→</td>
<td></td>
<td>AAA</td>
</tr>
</tbody>
</table>

Hu, et al. [Page 35]
The DHCP dual stack access includes three sets of Update_Request / Update_Response exchanges to create/update DHCPv4/v6 subscriber session.

1. Create a 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 a 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:

<table>
<thead>
<tr>
<th>RG</th>
<th>UP</th>
<th>CP</th>
<th>AAA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Static Subscriber</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Detection Req.</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>&lt;------to UP via Ci--</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Static Subscriber</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Detection Rep.</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>------to UP via Ci----&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>ARP/ND (REQ)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.1 &lt;-----</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>ARP/ND (ACK)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.2</td>
<td>---------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Relay the ARP/ND</td>
<td>AAA</td>
<td></td>
</tr>
<tr>
<td>3.3</td>
<td>------to CP via Si----&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>AAA</td>
<td></td>
</tr>
<tr>
<td>3.4</td>
<td>&lt;---------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Create subscriber</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>session Request</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Hu, et al [Page 37]
Figure 19: L2 Static Subscriber Access

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

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

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

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

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

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

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

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

Hu, et al                                                      [Page 39]
\[\text{<Update\_Response\ Message> := <Common Header> <Update\_Response\ TLV>}\]

5.2. PPPoE

5.2.1. IPv4 PPPoE Access

The following figure shows the IPv4 PPPoE access call flow.

<table>
<thead>
<tr>
<th></th>
<th>RG</th>
<th>UP</th>
<th>CP</th>
<th>AAA</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>PPPoE Disc</td>
<td>&lt;------------------via Si-------&gt;</td>
<td>PPPoE Disc</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>PPP LCP</td>
<td>&lt;------------------via Si-------&gt;</td>
<td>PPP LCP</td>
<td>AAA</td>
</tr>
<tr>
<td>3</td>
<td>PPP PAP/CHAP</td>
<td>&lt;------------------via Si-------&gt;</td>
<td>PPP PAP/CHAP</td>
<td>&lt;---------------------------&gt;</td>
</tr>
<tr>
<td>4</td>
<td>PPP IPCP</td>
<td>&lt;------------------via Si-------&gt;</td>
<td>PPP IPCP</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td>Create subscriber session Request</td>
<td>&lt;------------------via Ci-------&gt;</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td>Create subscriber session Response</td>
<td>&lt;------------------via Ci-------&gt;</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td>Accounting</td>
<td>&lt;---------------------------&gt;</td>
</tr>
</tbody>
</table>

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.

+-----------------+-----------------+-----------------+-----------------+-----------------+  
| RG              | UP              | CP              | AAA             |  
| PPPoE Disc      | PPPoE Disc      |                 |                 |  
| 1<--------------|<-------------->|<--------via Si-->|                 |  
| PPP LCP         | PPP LCP         |                 | AAA             |  
| 2<--------------|<-------------->|<--------via Si-->|<-------------->  |  
| PPP PAP/CHAP    | PPP PAP/CHAP    | AAA             |                 |  
| 3<--------------|<-------------->|<--------via Si-->|<-------------->  |  
| PPP IP6CP       | PPP IP6CP       |                 |                 |  
| 4<--------------|<-------------->|<--------via IP6CP|                 |  
|                 | Create subscriber session Request |                 |                 |  
| 5<-------------->|<--------via CI----->|                 |                 |  
|                 | Create subscriber session Response |                 |                 |  
| 6<-------------->|<--------via CI----->|                 |                 |  
| ND Negotiation  | ND Negotiation  |                 |                 |  
| 7<--------------|<-------------->|<--------via SI-->|                 |  
|                 | Update subscriber session Request |                 |                 |  
| 8<-------------->|<--------via CI----->|                 |                 |  
|                 | Update subscriber session Response |                 |                 |  
| 9<-------------->|<--------via CI----->|                 |                 |  

Hu, et al

[Page 41]
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:

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

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

Then, the RG will initialize an ND/DHCPv6 negotiation process with the CP (see 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>
  <IPv4 Subscriber TLV>
  <IPv4 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.

```
<table>
<thead>
<tr>
<th>RG</th>
<th>UP</th>
<th>CP</th>
<th>AAA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PPPoE Discovery</td>
<td>PPPoE Discovery</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>&lt;---------&gt;</td>
<td>&lt;--------via Si--------&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>PPP LCP</td>
<td>PPP LCP</td>
<td>AAA</td>
</tr>
<tr>
<td>2</td>
<td>&lt;---------&gt;</td>
<td>&lt;--------via Si--------&gt;</td>
<td>Req/Rep</td>
</tr>
<tr>
<td></td>
<td>PPP PAP/CHAP</td>
<td>PPP PAP/CHAP</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>&lt;---------&gt;</td>
<td>&lt;--------via Si--------&gt;</td>
<td>&lt;---------&gt;</td>
</tr>
<tr>
<td></td>
<td>PPP IPCP</td>
<td>PPP IPCP</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>&lt;---------&gt;</td>
<td>&lt;--------via Si--------&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Create v4 subscriber session Request</td>
</tr>
<tr>
<td>5</td>
<td>&lt;--------via Ci--------&gt;</td>
<td>Create v4 subscriber session Response</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Accounting</td>
</tr>
<tr>
<td>6</td>
<td>&lt;--------via Ci--------&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4'</td>
<td>&lt;---------&gt;</td>
<td>&lt;--------via Si--------&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>PPP IP6CP</td>
<td>PPP IP6CP</td>
<td></td>
</tr>
<tr>
<td>5'</td>
<td>&lt;--------via Ci--------&gt;</td>
<td>Create v6 subscriber session Request</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6'</td>
<td>&lt;--------via Ci--------&gt;</td>
<td>Create v6 subscriber session Response</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>ND Negotiation</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>ND Negotiation</td>
<td></td>
</tr>
</tbody>
</table>
```
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′)

\[\text{<Update Request Message> ::= <Common Header>}
\quad \text{<Basic Subscriber TLV>}
\quad \text{<PPP Subscriber TLV>}
\quad \text{<IPv6 Subscriber TLV>}
\quad \text{<IPv6 Routing TLV>}
\quad \text{[<Subscriber Policy TLV>]}\]

\[\text{<Update Response Message> ::= <Common Header>}
\quad \text{<Update Response TLV>}\]

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

\[\text{<Update Request Message> ::= <Common Header>}
\quad \text{<Basic Subscriber TLV>}
\quad \text{<PPP Subscriber TLV>}
\quad \text{<IPv6 Subscriber TLV>}
\quad \text{<IPv6 Routing TLV>}
\quad \text{[<Subscriber Policy TLV>]}\]

\[\text{<Update Response Message> ::= <Common Header>}
\quad \text{<Update Response TLV>}\]

5.3. WLAN Access

The following figure shows the WLAN access call flow.

<table>
<thead>
<tr>
<th>RG</th>
<th>UP</th>
<th>CP</th>
<th>AAA</th>
<th>WEB Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>DHCP</td>
<td>Discovery</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1&lt;-------</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DHCP</td>
<td>Discovery</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2&lt;-----via Si-----</td>
<td>AAA</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DHCP</td>
<td>Offer</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3&lt;--------via Si-----</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DHCP</td>
<td>Offer</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4&lt;--------</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DHCP</td>
<td>Request</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5--------&gt;</td>
<td>DHCP Request</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6--------via Si-----</td>
<td>Create session</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Hu, et al [Page 45]
WLAN access starts with the DHCP dial-up process (steps 1-6), after that the CP will create a subscriber session on the UP (steps 7-8). The formats of the session creation messages are as follows:
IPv4 Case:
<Update_Request Message> ::= <Common Header>
    <Basic Subscriber TLV>
    <IPv4 Subscriber TLV>
    <IPv4 Routing TLV>
    [<Subscriber Policy TLV>]

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

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

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

<table>
<thead>
<tr>
<th>RG</th>
<th>UP (LAC)</th>
<th>CP (LAC)</th>
<th>AAA</th>
<th>LNS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PPPoE</td>
<td>PPPoE</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Discovery</td>
<td>Discovery</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>&lt;---------</td>
<td>&lt;---via Si---&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PPP LCP</td>
<td>PPP LCP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>&lt;---------</td>
<td>&lt;---via Si---&gt;</td>
<td></td>
<td>AAA</td>
</tr>
<tr>
<td></td>
<td>PPP PAP/CHAP</td>
<td>PPP PAP/CHAP</td>
<td>Req/Rep</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>&lt;---------</td>
<td>&lt;---via Si---&gt;</td>
<td>&lt;------</td>
<td></td>
</tr>
<tr>
<td></td>
<td>PPP IPCP</td>
<td>PPP IPCP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>&lt;---------</td>
<td>&lt;---via Si---&gt;</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

```
P2P tunnel negotiation  
SCCRQ/  
SCCRP/  
SCCCN  
5 <---via Si--->  
/\  
| forward  
/\  
<-------via routing-------->
```

```
L2TP session negotiation  
ICRQ/  
ICRP/  
ICCN  
6 <---via Si--->  
/\  
| forward  
/\  
<-------via routing-------->
```

```
Create  
Subscriber session  
Request  
7 <---via Ci----  
Create  
```
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:

\[
\text{Update Request Message} ::= \text{Common Header} \\
\text{Basic Subscriber TLV} \\
\text{L2TP-LAC Subscriber TLV} \\
\text{L2TP-LAC Tunnel TLV} \\
\]

\[
\text{Update Response Message} ::= \text{Common Header} \\
\text{Update Response TLV} \\
\]

5.4.2. L2TP LNS IPv4 Access

<table>
<thead>
<tr>
<th>RG</th>
<th>LAC</th>
<th>UP(LNS)</th>
<th>AAA</th>
<th>CP(LNS)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PPPoE Discovery</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>&lt;------------&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PPP LCP</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>&lt;------------&gt;</td>
<td>AAA</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PPP PAP/CHAP</td>
<td></td>
<td>Req/Rep</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>&lt;------------&gt;</td>
<td>&lt;-----------------&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>L2TP tunnel negotiation</td>
<td>L2TP tunnel negotiation</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SCCRQ/</td>
<td>SCCRQ/</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SCCRP/</td>
<td>SCCRP/</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SCCCN</td>
<td>SCCCN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>&lt;------------&gt;</td>
<td>&lt;-----via Si-----&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>L2TP session</td>
<td>L2TP session</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
In this case, the BNG is running as an LNS and separated into LNS-CP and LNS-UP. Steps 1-5 finish the normal L2TP dial-up process. When the L2TP session and tunnel negotiations are finished, the LNS-CP will create an L2TP LNS subscriber session on the LNS-UP. The format of the messages is as follows:

![Diagram](image-url)

Figure 25: IPv4 L2TP-LNS Access
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:

<table>
<thead>
<tr>
<th>RG</th>
<th>LAC</th>
<th>UP (LNS)</th>
<th>AAA</th>
<th>CP (LNS)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>PPPoE</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Discovery</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>&lt;----</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>PPP LCP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>&lt;----</td>
<td>AAA</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>PPP PAP/CHAP</td>
<td>Reg/Rep</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>&lt;----</td>
<td>&lt;-------------------&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>L2TP tunnel</td>
<td>L2TP tunnel</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>negotiation</td>
<td>negotiation</td>
<td></td>
</tr>
</tbody>
</table>

5.4.3. L2TP LNS IPv6 Access
<table>
<thead>
<tr>
<th></th>
<th>SCCRQ/</th>
<th>SCCRQ/</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SCCRP/</td>
<td>SCCRP/</td>
</tr>
<tr>
<td></td>
<td>SCCCN</td>
<td>SCCCN</td>
</tr>
</tbody>
</table>
| 4|<------------|<------via Si------>
|   | L2TP session | L2TP session |
|   | negotiation | negotiation |
|   | ICRQ/ | ICRQ/ |
|   | ICRP/ | ICRP/ |
|   | ICCN  | ICCN  |
| 5|<------------|<------via Si------>
| Create | subscriber session Request |
| 6|------via Ci------>
| Create | subscriber session Response |
| 7|------via Ci------>
| PAP/CHAP (Triggered by LNS) |
| 8|--------------------------------------------|
| AAA | Req/Rep |
| 9|<-------->
| PPP IP6CP |
| 10|--------------------------------------------|
| Update | subscriber session Request |
| 11|------via Ci------>
| Update | subscriber session Response |
| 12|------via Ci------>
| ND negotiation | ND negotiation |
| 13|--------------------------------------------|
| Update |
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:

\[
\text{<Update Request Message> ::= <Common Header>}
\text{<L2TP-LNS Subscriber TLV>}
\text{<Basic Subscriber TLV>}
\text{<PPP Subscriber TLV>}
\text{<IPv6 Subscriber TLV>}
\text{<IPv6 Routing TLV>}
\text{<L2TP-LNS Tunnel TLV>}
\text{[<Subscriber Policy TLV>]}
\]

\[
\text{<Update Response Message> ::= <Common Header>}
\text{<Update Response TLV>}
\]

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:

\[
\text{<Update Request Message> ::= <Common Header>}
\text{<L2TP-LNS Subscriber TLV>}
\text{<Basic Subscriber TLV>}
\text{<PPP Subscriber TLV>}
\text{<IPv6 Subscriber TLV>}
\text{<IPv6 Routing TLV>}
\text{<L2TP-LNS Tunnel TLV>}
\text{[<Subscriber Policy TLV>]}
\]

\[
\text{<Update Response Message> ::= <Common Header>}
\text{<Update Response TLV>}
\]

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:

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

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

5.5. CGN (Carrier Grade NAT)

<table>
<thead>
<tr>
<th>RG</th>
<th>UP</th>
<th>CP</th>
<th>AAA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Public Address Block Allocation Request</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>&lt;--------via Ci------&gt;</td>
<td>Public Address Block Allocation Reply</td>
<td></td>
</tr>
<tr>
<td></td>
<td>------------------</td>
<td>------------------</td>
<td>-----------</td>
</tr>
<tr>
<td></td>
<td>Subscriber access request</td>
<td>Subscriber access request</td>
<td>AAA Reg/Rep</td>
</tr>
<tr>
<td>3</td>
<td>&lt;--------------&gt;</td>
<td>4</td>
<td>5&lt;----------</td>
</tr>
<tr>
<td></td>
<td>1&lt;---------via Ci---------&gt;</td>
<td>2&lt;---------via Ci---------&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Public Address Block Allocation Request</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>&lt;--------via Ci---------&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Create subscriber session Request</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>&lt;--------via Ci---------&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Create subscriber session Response (with NAT information)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>&lt;--------via Ci---------&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Accounting with source information</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>&lt;--------------&gt;</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

Hu, et al

[Page 54]
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.

\[\text{<Addr\_Allocation\_Req Message> ::= <Common Header>}
\text{<Request Address Allocation TLV>}\]

\[\text{<Addr\_Allocation\_Ack Message> ::= <Common Header>}
\text{<Address Assignment Response TLV>}\]

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

<table>
<thead>
<tr>
<th></th>
<th>RG</th>
<th>UP</th>
<th>CP</th>
<th>AAA</th>
<th>WEB Server</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>User</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>User traffic</td>
<td>1 ----------</td>
<td>User</td>
<td></td>
<td>AAA Req/Rep</td>
<td></td>
</tr>
<tr>
<td>User traffic</td>
<td>2 --------via Si--------&gt;</td>
<td>AAA</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Create session Request</td>
<td>3 &lt;------------&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Create session</td>
<td>4 &lt;--------via Ci------&gt;</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
In this case, IP traffic from the RG will trigger the CP to authenticate the RG by checking the source IP and the exchanges with the AAA server. Once the RG passed the authentication, the CP will create a corresponding subscriber session on the UP through the following message exchanges:

**IPv4 Case:**

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

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

Figure 28: Web Authentication based L3 Leased Line Access
IPv6 Case:
<Update_Request Message> ::= <Common Header>
<Basic Subscriber TLV>
<IPv6 Subscriber TLV>
<IPv6 Routing TLV>
[<Subscriber Policy TLV>]

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

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

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

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

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

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

5.6.2. User Traffic Trigger

<table>
<thead>
<tr>
<th>RG</th>
<th>UP</th>
<th>CP</th>
<th>AAA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>L3 access control list</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 &lt;--via Ci-----</td>
<td></td>
</tr>
<tr>
<td>User</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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:

\[
\text{<Update Request Message> ::= <Common Header>}
\text{<Basic Subscriber TLV>}
\text{<IPv4 Subscriber TLV>}
\text{<IPv4 Routing TLV>}
[<Subscriber Policy TLV>]
\]

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

IPv6 Case:

\[
\text{<Update Request Message> ::= <Common Header>}
\text{<Basic Subscriber TLV>}
\text{<IPv6 Subscriber TLV>}
\text{<IPv6 Routing TLV>}
[<Subscriber Policy TLV>]
\]

\[
\text{<Update Response Message> ::= <Common Header>}
\text{<Update Response TLV>}
\]
5.7. Multicast Service Access

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

IPv4 Case:

```plaintext
<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:

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

Figure 30: Multicast Access
<Update_Response Message> ::= <Common Header>
    <Update Response TLV>
6. S-CUSP Message Formats

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

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

This section specifies the format of the common S-CUSP message header and lists the defined messages.

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

6.1. Common Message Header

S-CUSP Common Message Header:

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Ver  |  Resv | Message-Type  |        Message-Length         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            Reserved           |        Transaction-ID         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 6.1: S-CUSP Message Common Header

- **Ver (4 bits):** The major version of the protocol. This document specifies version 1. Different major versions of the protocol may have significantly different message 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.

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

- **Message-Type (8 bits):** The set of message types specified in this document is listed in Section 9.1.

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

<table>
<thead>
<tr>
<th>Type</th>
<th>Name</th>
<th>Notes and TLVs that can be carried</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Hello</td>
<td>Hello TLV, Keepalive TLV.</td>
</tr>
<tr>
<td>2</td>
<td>Keepalive</td>
<td>A common header with the Keepalive message type.</td>
</tr>
<tr>
<td>3</td>
<td>Sync_Request</td>
<td>Synchronization request.</td>
</tr>
<tr>
<td>4</td>
<td>Sync_Begin</td>
<td>Synchronization starts.</td>
</tr>
<tr>
<td>5</td>
<td>Sync_Data</td>
<td>Synchronization data: TLVs specified in Section 5.</td>
</tr>
<tr>
<td>6</td>
<td>Sync_End</td>
<td>End synchronization.</td>
</tr>
<tr>
<td>7</td>
<td>Update_Request</td>
<td>TLVs specified in Sections 7.6-7.9.</td>
</tr>
<tr>
<td>8</td>
<td>Update_Response</td>
<td>TLVs specified in Sections 7.6-7.9.</td>
</tr>
</tbody>
</table>

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

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

The format of the Keepalive message is as follows:

<Keepalive Message> ::= <Common Header>

6.2.3. Sync_Request Message

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

The format of the Sync_Request message is as follows:

<Sync_Request Message> ::= <Common Header>

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

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

The format of Sync_Begin message is as follows:

\[
\text{<Sync\_Begin Message>} ::= \text{<Common Header>}
\text{<Error Information TLV>}
\]

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.

\[
\text{<Sync\_Data Message>} ::= \text{<Common Header>}
\text{[<Resource Reporting TLVs>]}
\]

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

\[
\text{<Sync\_Data Message>} ::= \text{<Common Header>}
\text{[<User Routing TLVs>]}
\]
6.2.6. Sync_End Message

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

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

The return/error codes are listed as follows:

0: Success, synchronization finished.
1: Failure, malformed message received.
2: One or more of the TLVs was not understood.

6.2.7. Update_Request Message

The Update_Request message is a multi-purpose 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 is a request to delete a corresponding session.

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>
[<User Traffic Statistics Report TLV>]
[<User Detection Result Report TLV>]

6.4. Report Message

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

<table>
<thead>
<tr>
<th>Type</th>
<th>Message Name</th>
<th>TLV that is carried</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>Addr_Allocation_Req</td>
<td>Address Allocation Request</td>
</tr>
<tr>
<td>201</td>
<td>Addr_Allocation_Ack</td>
<td>Address Allocation Response</td>
</tr>
<tr>
<td>202</td>
<td>Addr_Renew_Req</td>
<td>Address Renewal Request</td>
</tr>
<tr>
<td>203</td>
<td>Addr_Renew_Ack</td>
<td>Address Renewal Response</td>
</tr>
<tr>
<td>204</td>
<td>Addr_Release_Req</td>
<td>Address Release Request</td>
</tr>
<tr>
<td>205</td>
<td>Addr_Release_Ack</td>
<td>Address Release Response</td>
</tr>
</tbody>
</table>

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_Req message is as follows:

<Addr_Renew_Ack Message> ::= <Common Header>
                           <Address Renewal Response TLV>

6.5.5.  Addr_Release_Req Message

The Addr_Release_Req message is used to request address release. The format of 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, in conjunction with the vendor TLV and vendor sub-TLV, can be used by vendors to extend the S-CUSP protocol. The message type is 11. If the receiver does not recognize the message, an Error message will be returned to the sender.

The format of the Vendor message is as follows:

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

6.7 Error Message

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

The format of the Error message is as below:

>Error Message> ::= <Common Header>
  <Error Information TLV>
7. S-CUSP TLVs and Sub-TLVs

This section specifies the following:

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

See Section 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.

0                   1                   2                   3
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Oper  |      TLV-Type         |       TLV-Length              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                             Value                             ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

Figure 32: Common TLV Header

- Oper (4 bits): For Message-Types that specify an operation on a data set, the Oper field is interpreted as Update, Delete, or Reserved as specified in Section 9.3. For all other Message-Types, the Oper field MUST be sent as zero and ignored on receipt.

- TLV-Type (12 bits): The Type of a TLV. TLV-Type specifies the interpretation and format of the Value field of the TLV. See Section 9.2.

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

- Value (variable length): This is the 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.

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

- **MAC-Addr**: 6 octets. Ethernet MAC Address [RFC7042].

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

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

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

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


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:

```
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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              Type             |          Length               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Value                                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-...
```

**Figure 33: Sub-TLV Header**

- **Type (2 bytes):** The Type of a Sub-TLV. The Type field specified the interpretation and format of the Value field of the TLV. Sub-TLV Type values have the same meaning regardless of the TLV Type of the TLV within which the sub-TLV occurs. See Section 9.4.

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

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

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

7.3.1. Name sub-TLVs

This document defines the following name sub-TLVs that are used to carry the name of the corresponding object. The length of each of these sub-TLVs is variable from 1 to 255 octets. The value is of type STRING padded with zeros octets to 4-octet alignment.

<table>
<thead>
<tr>
<th>Type</th>
<th>Sub-TLV Name</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>VRF-Name</td>
<td>The name of a VRF</td>
</tr>
<tr>
<td>2</td>
<td>Ingress-QoS-Profile</td>
<td>The name of an ingress QoS profile</td>
</tr>
<tr>
<td>3</td>
<td>Egress-QoS-Profile</td>
<td>The name of an egress QoS profile</td>
</tr>
<tr>
<td>4</td>
<td>User-ACL-Policy</td>
<td>The name of an ACL policy</td>
</tr>
<tr>
<td>5</td>
<td>Multicast-ProfileV4</td>
<td>The name of an IPv4 multicast profile</td>
</tr>
<tr>
<td>6</td>
<td>Multicast-ProfileV6</td>
<td>The name of an IPv6 multicast profile</td>
</tr>
<tr>
<td>7</td>
<td>NAT-Instance</td>
<td>The name of a NAT instance</td>
</tr>
<tr>
<td>8</td>
<td>Pool-Name</td>
<td>The name of an address pool</td>
</tr>
</tbody>
</table>
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. The sub-TLV length is 16. The format is as shown in Figure 34.

```
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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  CIR (Committed Information Rate)             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  PIR (Peak Information Rate)                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  CBS (Committed Burst Size)                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  PBS (Peak Burst Size)                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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. Its sub-TLV length is 16 octets. The format of the value part is as defined below.
Figure 35: Egress-CAR sub-TLV

Where:

- **CIR** (4 bytes): Guaranteed rate in bits/second.
- **PIR** (4 bytes): Burst rate in bits/second.
- **CBS** (4 bytes): The token bucket in bytes.
- **PBS** (4 bytes): Burst token bucket in bytes.

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

### 7.3.4. If-Desc sub-TLV

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

The sub-TLV type is 11. The sub-TLV length is 12 octets. The format depends on the If-Type. The format of the value part is as follows:
If-Type: 8 bits in length. The value of this field indicates the type of an interface. The If-Type values defined in this document are listed in Section 9.6.

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

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

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

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

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

Logic-ID (32 bits): An identifier of a virtual interface (e.g., If-Type: 6-7)
7.3.5. IPv6 Address List sub-TLV

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

The format of IPv6 Addresses sub-TLV is as follows:

```
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
\~ IPv6 Address \~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
\~ IPv6 Address \~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
\~ ... \~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
\~ IPv6 Address \~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

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:

```
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
| Vendor ID |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
| Sub-Type | Sub-Type-Version |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
\~ value (other as specified by vendor) \~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

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 desirable for a vendor to be able to define multiple different vendor sub-TLVs and to keep track of different versions of its vendor-defined sub-TLVs. Thus, it is RECOMMENDED that the vendor assign a Sub-Type value for each of that vendor’s sub-TLVs that is different from other Sub-Type values that vendor has used. Also, when modifying a vendor-defined sub-TLV in a way potentially incompatible with a previous definition, the vendor SHOULD increase the value it is using in the Sub-Type-Version field.

7.4. The Hello TLV

The Hello TLV is defined to be carried in the Hello message for version and capabilities negotiation. It indicates the S-CUSP sub-version and capabilities supported. The format of Hello 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          VerSupported                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           Vendor ID                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Capabilities                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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. It 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, so implementations of the version specified herein will set this field to zero. The Capabilities field of the Hello TLV MUST be checked before any other TLVs in the Hello because capabilities defined in the future might extend existing TLVs or permit new TLVs.

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

The Keepalive TLV is carried in the Hello message. It provides timing information for this feature. The format of Hello 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Keepalive   | DeadTimer     |            Reserved           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 40: Keepalive TLV

Where:

The TLV type: 102.

The value of the Length field is 4 octets.

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

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

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:

```
+--------+--------+--------+--------+
| Message-Type | Reserved | TLV-Type | Error Code |
+--------+--------+--------+--------+
```

Figure 41: Error Information TLV

Where:

- The TLV type: 101.
- The value of the Length field is 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 on a UP.
The format of the BAS Function TLV value part 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 2 3 4 5 6 7 8 9 0 1 |
+-------------------+-------------------+-------------------+-------------------+
|                      |                      |                      |                      |
+-------------------+-------------------+-------------------+-------------------+
| 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 2 3 4 5 6 7 8 9 0 1 |
+-------------------+-------------------+-------------------+-------------------+
|                      |                      |                      |                      |
+-------------------+-------------------+-------------------+-------------------+
| If-Index           | If-Index           | If-Index           | If-Index           |
+-------------------+-------------------+-------------------+-------------------+
| Access-Mode       | Auth-Method4      | Auth-Method6      | Reserved           |
+-------------------+-------------------+-------------------+-------------------+
| Flags             | Flags             | Flags             | Flags             |
+-------------------+-------------------+-------------------+-------------------+
| sub-TLVs (optional) | sub-TLVs (optional) | sub-TLVs (optional) | sub-TLVs (optional) |
+-------------------+-------------------+-------------------+-------------------+
```

Figure 42: BAS Function TLV

Where:

The TLV type: 1.

The value of the Length field is variable.

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

Access-Mode: 1 byte in length. It indicates the access mode of the interface. The defined values are listed in Section 9.7.

Auth-Method4: 1 byte in length. It indicates the authentication on this interface for the IPv4 scenario. This field is defined as a bitmap. The bits defined in this document are listed in Section 9.8. Other bits are reserved and MUST be sent as zero and ignored on receipt.

Auth-Method6: 1 byte in length. It indicates the authentication on this interface for the IPv6 scenario. This field is defined as a bitmap. The bits defined in this document are listed in Section 9.8. Other bits are reserved and MUST be sent as zero and ignored on receipt.

sub-TLVs:
- The IF-Desc sub-TLV can be carried.
- If-Desc sub-TLV: carries the interface information.
The flags field is defined 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 |
```

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

Typically, after an S-CUSP session is established between a UP and a CP, the CP will allocate one or more blocks of IP addresses to the UP. Those IP addresses will be allocated to subscribers who will
dial-up (as defined in Section 2.1) to the UP. To make sure that other nodes within the network learn how to reach those IP addresses, the CP needs to install one or more routes that can reach those IP addresses on the UP and notify the UP to advertise the routes to the network.

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

7.8.1. IPv4 Routing TLV

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

The format of the TLV value part is as below:

```
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          User ID                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Dest-Address                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Next-Hop                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Out-If-Index                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Cost                                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Tag                                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        Route Type             |          Reserved           |A|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                          sub-TLVs  (optional)                 ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 44: IPv4 Routing TLV

Where:

The TLV Type: 7

The TLV Length: Variable

User-ID: 4 bytes in length. This field carries the user identifier. It is filled with all Fs when a non-user route is delivered to the UP.
Dest-Address (IPv4-Address type): Identifies the destination address.

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

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

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

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

Route-Type (2 bytes): The value of this field indicates the route type. The values defined in this document are listed in Section 9.9.

Advertise-Flag: 1 bit shown as "A" is the figure above. Indicates whether the IP should advertise the route. The following flag values are defined:

- 0: Not advertised,
- 1: Advertised.

sub-TLVs: The VRF-Name and/or If-Desc sub-TLVs can be carried.
- VRF-Name sub-TLV: indicates which VRF the route belongs to.
- If-Desc sub-TLV: carries the interface information.

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:

```
+---------------------------------------------------------------+
|                          User ID                              |
+---------------------------------------------------------------+
|                          IPv6 Dest-Address                    |
+---------------------------------------------------------------+
|                          IPv6 Next-Hop                        |
+---------------------------------------------------------------+
|                          Out-If-Index                         |
+---------------------------------------------------------------+
|                          Cost                                 |
+---------------------------------------------------------------+
|                          Tag                                  |
+---------------------------------------------------------------+
|        Route Type             |          Reserved           |A|                              |
+---------------------------------------------------------------+
|                          sub-TLVs (optional)                  |
+---------------------------------------------------------------+
```

Figure 45: IPv6 Routing TLV

Where:

The TLV Type: 7

The TLV Length is Variable.

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

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

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

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

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

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

Route-Type: (2 bytes): The value of this field indicates the route type. The values defined in this document are listed in Section 9.9.
Advertise-Flag: 1 bit shown as "A" is the figure above. Indicates whether UP should advertise the route. 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:

```
+-----------------------------+-----------------------------+
| User ID                     | Session ID                  |
+-----------------------------+-----------------------------+
| User MAC                    | User MAC                    |
+-----------------------------+-----------------------------+
| Access Type                 | Access Type                 |
| Sub-access Type             | Sub-access Type             |
+-----------------------------+-----------------------------+
| Account Type                | Account Type                |
| Address Family              | Address Family              |
+-----------------------------+-----------------------------+
| C-VID                       | P-VID                       |
+-----------------------------+-----------------------------+
| Detect Times                | Detect Times                |
+-----------------------------+-----------------------------+
| If-Index                    | If-Index                    |
+-----------------------------+-----------------------------+
~ sub-TLVs (optional) ~
+-----------------------------+-----------------------------+
```

Figure 46: Basic Subscriber TLV

Where:

- The TLV Type: 2.
- The TLV is variable in length.
- User-ID (4 bytes): The identifier of a subscriber.
- Session-ID (4 bytes): Session ID of a PPPoE subscriber. The value zero identifies a non-PPPoE subscriber.
- User-Mac (MAC-Addr type): The MAC Address of a subscriber.
- Oper-ID (1 byte): Indicates the ID of an operation performed by a user. This field is carried in the response from the UP.
- Reserved (1 byte): MUST be sent as zero and ignored on receipt.
- Access-Type (1 byte): Indicates the type of subscriber access. Values defined in this document are listed in Section 9.10.
Sub-Access-Type (1 byte): Indicates whether PPP termination or PPP relay is used.
   0: Reserved
   1: PPP Relay (for LAC)
   2: PPP termination (for LNS)

Account-Type (1 byte):
   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:

```
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----------------------------------------------+
|                                               |
|                                               |
|                                               |
|                                               |
|                                               |
|                                               |
|                                               |
|                                               |
|                                               |
+-----------------------------------------------+
```

Figure 47: PPP Subscriber TLV

Where:

- The TLV type: 3.
- The TLV length is 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:

```
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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          User ID                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          User IPv4 Address                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Gateway IPv4 Address                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          MTU                  |   Reserved            |U|E|W|P|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                          VRF-Name sub-TLV                     ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 48: IPv4 Subscriber TLV

Where:

The TLV type: 4.

The TLV length is 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.


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:

```
+---------------------------------------------------------------+
|                          User-ID                              |
+---------------------------------------------------------------+
~ User PD-Address (IPv6 Address List sub-TLV) ~
~ Gateway ND-Address (IPv6 Address List sub-TLV) ~
|                          User Link-Local-Address              |
|                          IPv6 Interface ID                    |
|                          IPv6 Interface ID (cont.)            |
|          MTU                  |   Reserved            |U|E|W|P|
| VRF Name sub-TLV (optional) ~
+---------------------------------------------------------------+
```

Figure 49: IPv6 Subscriber TLV

Where:

- The TLV type: 5.
- The TLV length is variable.
- User-ID (4 bytes): The identifier of a subscriber.
- User PD-Addresses (IPv6 Address List): Carries a list of Prefix Delegation (PD) addresses of the subscriber.
- User ND-Addresses (IPv6 Address List): Carries a list of Neighbor Discovery (ND) addresses of the subscriber.
- User Link-Local-Address (IPv6-Address): The link-local address of the subscriber.
- IPv6 Interface ID (8 bytes): The identifier of an IPv6 interface.
- Portal Force 1 bit (P): Push advertisement, 0: off, 1: on.
Web-Force 1 bit (W): Push IPv6 Web, 0: off, 1: on.

Echo-Enable 1 bit (E): The UP returns ARP Req or PPP Echo. 0: off; 1: on.

IPv6-URPF 1 bit (U): User Reverse Path Forwarding (URPF) flag, 0: off; 1: on.

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

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

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:

```
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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          If-Index                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          C-VID               |           P-VID               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          User Address                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Gateway Address                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          User MAC                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          User MAC (cont.)                   |           Reserved            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                       sub-TLVs (optional)                     ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 50: IPv4 Static Subscriber TLV

Where:

The TLV type: 6.
The TLV length is 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:

```
+-------------------------------------------+
<table>
<thead>
<tr>
<th>If-Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>C-VID</td>
</tr>
<tr>
<td>-------------------------------------------</td>
</tr>
<tr>
<td>User Address</td>
</tr>
<tr>
<td>-------------------------------------------</td>
</tr>
<tr>
<td>Gateway Address</td>
</tr>
<tr>
<td>-------------------------------------------</td>
</tr>
<tr>
<td>User MAC</td>
</tr>
<tr>
<td>-------------------------------------------</td>
</tr>
<tr>
<td>User MAC (cont.)</td>
</tr>
<tr>
<td>-------------------------------------------</td>
</tr>
<tr>
<td>sub-TLVs (optional)</td>
</tr>
</tbody>
</table>
+-------------------------------------------+
```

Figure 51: IPv6 Static Subscriber Detect TLV

Where:

- The TLV type: 6.
- The TLV length is 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 an L2TP LAC access subscriber. It will be carried in an Update_Request message when L2TP LAC access is used.

The format of the TLV value part is as follows:

```
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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          User ID                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Local Tunnel ID          |     Local Session ID          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Remote Tunnel ID         |     Remote Session ID         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 52: L2TP-LAC Subscriber TLV

Where:

The TLV type: 11.

The TLV is 12 octets long.

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

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

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

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

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

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:

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          User ID                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Local Tunnel ID          |     Local Session ID          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Remote Tunnel ID         |     Remote Session ID         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 53: L2TP-LNS Subscriber TLV

Where:

- The TLV type: 12.
- The TLV length is 12 octets.
- User-ID (4 bytes): The identifier of a user/subscriber.
- Local-Tunnel-ID (2 bytes): The local ID of the L2TP tunnel.
- Local-Session-ID (2 bytes): The local session ID with the L2TP tunnel.
- Remote-Tunnel-ID (2 bytes): The identifier of the L2TP tunnel at the remote end-point.
- Remote-Session-ID (2 bytes): The session ID of the L2TP tunnel at the remote end-point.

### 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:

```
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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       Local Tunnel ID         |       Remote Tunnel ID        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Source Port         |        Destination Port       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                          Tunnel Source Address                ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                          Tunnel Destination Address           ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                          VRF Name sub-TLV                     ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 54: L2TP-LAC Tunnel TLV

Where:

The TLV type: 13.

The TLV length is 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:

```
  0 | 1 | 2 | 3
  +---+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     | Local Tunnel ID | Remote Tunnel ID |
  +---+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     | Source Port | Destination Port |
  +---+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     ~ Tunnel Source Address ~
  +---+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     ~ Tunnel Destination Address ~
  +---+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     ~ VRF Name sub-TLV ~
  +---+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 55: L2TP-LNS Tunnel TLV

Where:

- The TLV type: 14.
- The TLV length is 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:

```
                  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
          +-----------------------------------------------+
          | User-ID                                         |
          +-----------------------------------------------+
          | User-Trans-ID | Oper-Code | Oper-Result | Reserved |
          +-----------------------------------------------+
          | Error-Code                                          |
          +--------------------------------------------------+
```

Figure 56: Update Response TLV

Where:

- The TLV type: 302.
- The TLV length is 12.
- User-ID (4 bytes): A unique identifier of a user/subscriber.
- User-Trans-ID (1 byte): In the case of dual-stack access or when modifying a session, User-Trans-ID is used to identify a user operation transaction. It is used by the CP to correlate a response to a specific request.
- Oper-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:

```
+-----------------------------------------------+
|                          User ID                              |
+-----------------------------------------------+
|   I-Priority  |   E-Priority  |   Reserved                    |
+-----------------------------------------------+
~                          sub-TLVs                             ~
+-----------------------------------------------+
```

Figure 57: User QoS Policy Information TLV

Where:

The TLV type: 6.

The TLV length is variable.

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

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

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

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

Ingress-CAR sub-TLV: Upstream CAR.

Egress-CAR sub-TLV: Downstream CAR.

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

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

User-ACL-Policy Sub-TLV: All ACL user policies, including v4ACLIN, v4ACLOUT, v6ACLIN, v6ACLOUT, v4WEBACL, v6WEBACL, 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:

```
+--------------------------------------------------+
| User-ID                                           |
| +--------------------------------------------------+
| NAT-Port-Start | NAT-Port-End |
| +--------------------------------------------------+
| NAT-Address   |
| +--------------------------------------------------+
```

Figure 58: Subscriber CGN Port Range TLV

Where:

The TLV type: 15.

The TLV length is 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:

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>++++++++++++----------------------------------++</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>If-Index</td>
<td></td>
<td></td>
</tr>
<tr>
<td>+++++++------------------------------+++++++++++</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>MAC Address (upper part)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>+++++++----------------------------+++++++</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>MAC Address (lower part)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Phy-State</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Reserved</td>
<td></td>
<td></td>
</tr>
<tr>
<td>+++++++------------------------------++</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Phy-State</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Reserved</td>
<td></td>
<td></td>
</tr>
<tr>
<td>+++++++----------------------------+++++++</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>MTU</td>
<td></td>
<td></td>
</tr>
<tr>
<td>+++++++----------------------------+++++++</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>sub-TLVs (optional)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>+++++++----------------------------+++++++</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 59: Interface Status TLV

Where:

The TLV type: 200.

The TLV length is variable.

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

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

Phy-State (1 byte): Physical status of the interface. 0: down, 1: Up

MTU (4 bytes): Interface MTU value.

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

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

7.10.2. Board Status TLV

The Board Status TLV is used to carry the status information of a board on an UP. It is carried in a Report message.
The format of Board Status 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Board-Type  | Board-State   |   Reserved    |   Chassis     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               Slot            |           Reserved            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 60: Interface Resource TLV

Where:
- The TLV type: 201.
- The TLV length is 8 octets.
- Chassis (1 byte): The chassis number of the Board.
- Slot (1 byte): The slot number of the Board.
- Sub-Slot (1 byte): The sub-slot number of the Board.
- Board-Type (1 byte):
  1: CGN Service Process Unit (SPU) board.
  2: Line Process Unit (LPU) Board.
- Board-State (1 byte):
  0: Normal.
  1: Abnormal.

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

7.11. CGN TLVs

7.11.1. Address Allocation Request TLV

The Address Allocation Request TLV is used to request address allocation from CP. An address Pool-Name sub-TLV is carried to indicate from which address pool to allocate addresses. The Address Allocation Request TLV is carried in the Addr_Allocation_Req message.
The format of the value part of this TLV is as follows:

```
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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                          Pool-Name sub-TLV                    ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 61: Address Allocation Request TLV

Where:

The TLV type: 400.

The TLV length is 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:

```
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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Lease Time                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        IPv4 Addr and Mask                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        IPv4 Addr and Mask continued           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Error-Code                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                        Pool-Name sub-TLV                      ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 62: Address Assignment Response TLV

Where:

The TLV type: 401.

The TLV length is variable.
Lease Time (4 bytes): Duration of address lease in seconds.

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


  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:

```
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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     IPv4 Address and Mask                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     IPv4 Address and Mask continued           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                     Pool-Name sub-TLV                         ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 63: Address Renewal Request TLV

Where:

The TLV type: 402.

The TLV length is 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:

```
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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     IPv4 Address and Mask                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     IPv4 Address and Mask continued           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Error-Code                                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                     Pool-Name sub-TLV                         ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 64: Address Renewal Response TLV

Where:

The TLV type: 403.

The TLV length is variable.

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

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

0: Renew succeeded.

1: Renew failed.

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

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

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

3004 (Subnet-Conflict): Subnets in the address pool have been assigned to other clients.
Pool Name sub-TLV: A Pool-Name Sub-TLV to indicate from which Address pool to renew the address.

7.11.5. Address Release Request TLV

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

The value part of this TLV is formatted 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     IPv4 Address and Mask                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     IPv4 Address and Mask continued           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                     Pool-Name sub-TLV                         ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 65: Address Release Request TLV

Where:

- The TLV type: 404.
- The TLV length is 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:

```
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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     IPv4 Address and Mask                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     IPv4 Address and Mask continued           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Error-Code                                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                     Pool-Name sub-TLV                         ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 66: Address Renewal Response TLV

Where:

- The TLV type: 405.
- The TLV length is variable.
- **Client-IP (IPv4-Address type):** The released IPv4 address.
- **Error-Code (4 bytes):** Indicates success or an error.
  - 0: Address release success.
  - 1: Address release failed.
  - 3001 (Pool-Mismatch): The corresponding address pool cannot be found.
  - 3003 (Subnet-Mismatch): The address cannot be found.
  - 3004 (Subnet-Conflict): The address has been allocated to another subscriber.
- **Pool-Name sub-TLV:** A Pool-Name Sub-TLV to indicate from which address pool to release the address.

### 7.12. Event TLVs
7.12.1. Subscriber Traffic Statistics TLV

The Subscriber Traffic Statistics TLV is used to return the traffic statistics of a user/subscriber. The format of this 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                User-ID                                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                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 is 72 octets.

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

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

- 0: IPv4 traffic.
- 1: IPv6 traffic.
- 2: Dual stack traffic.

Ingress Packets (8 bytes): The number of the packets in 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).
The format of this TLV value part is as follows:

```
+---------------------------------+-----------------------------------+
|                          User-ID                           |                  |
| Detect Type   | Detect Result |          Reserved             |                  |
+---------------------------------+-----------------------------------+
```

Figure 68: Subscriber Detection Result TLV

Where:

The TLV type: 301.

The TLV length is 8 octets.

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

Detect-Type (1 byte):  

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

Detect-Result (1 byte):  

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:

```
+----------------+----------------+----------------+----------------+----------------+
| Vendor ID      | Sub-Type       | Sub-Type-Version|
+----------------+----------------+----------------+
```

Figure 69: Vendor TLV

Where:

- The TLV type: 1024.
- The TLV length is variable.
- Vendor-ID (4 bytes): Vendor ID as defined in RADIUS [RFC2865].
- Sub-Type (2 bytes): Used by the Vendor to distinguish multiple different vendor messages.
- Sub-Type-Version (2 bytes): Used by the Vendor to distinguish different versions of a Vendor-Defined message sub-type.
- Sub-TLVs (variable): Sub-TLVs as specified by the vendor.

Since Vendor code will be handling the TLV after the Vendor ID field is recognized, the remainder of the TLV 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 Series Editor (RSE), the Independent Submissions Editor (ISE), 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 Plane 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
8.3. EANTC Testing

EANTC (European Advanced Networking Test Center (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."

9. Tables of S-CUSP Codepoints

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

9.1. Message Types

<table>
<thead>
<tr>
<th>Type</th>
<th>Name</th>
<th>Section of this document</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>reserved</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Hello</td>
<td>6.2.1.</td>
</tr>
<tr>
<td>2</td>
<td>Keepalive</td>
<td>6.2.2.</td>
</tr>
<tr>
<td>3</td>
<td>Sync_Request</td>
<td>6.2.3.</td>
</tr>
<tr>
<td>4</td>
<td>Sync_Begin</td>
<td>6.2.4.</td>
</tr>
<tr>
<td>5</td>
<td>Sync_Data</td>
<td>6.2.5.</td>
</tr>
<tr>
<td>6</td>
<td>Sync_End</td>
<td>6.2.6.</td>
</tr>
<tr>
<td>7</td>
<td>Update_Request</td>
<td>6.2.7.</td>
</tr>
<tr>
<td>8</td>
<td>Update_Response</td>
<td>6.2.8.</td>
</tr>
<tr>
<td>9</td>
<td>Report</td>
<td>6.4.</td>
</tr>
<tr>
<td>10</td>
<td>Event</td>
<td>6.3.</td>
</tr>
<tr>
<td>11</td>
<td>Vendor</td>
<td>6.6.</td>
</tr>
<tr>
<td>12</td>
<td>Error</td>
<td>6.7.</td>
</tr>
<tr>
<td>13-199</td>
<td>unassigned</td>
<td></td>
</tr>
<tr>
<td>200</td>
<td>Addr_Allocation_Req</td>
<td>6.5.1.</td>
</tr>
<tr>
<td>201</td>
<td>Addr_Allocation_Ack</td>
<td>6.5.2.</td>
</tr>
<tr>
<td>202</td>
<td>Addr_Renew_Req</td>
<td>6.5.3.</td>
</tr>
<tr>
<td>203</td>
<td>Addr_Renew_Ack</td>
<td>6.5.4.</td>
</tr>
<tr>
<td>204</td>
<td>Addr_Release_Req</td>
<td>6.5.5.</td>
</tr>
<tr>
<td>205</td>
<td>Addr_Release_Ack</td>
<td>6.5.6.</td>
</tr>
<tr>
<td>206-254</td>
<td>unassigned</td>
<td></td>
</tr>
<tr>
<td>255</td>
<td>reserved</td>
<td></td>
</tr>
</tbody>
</table>

9.2. TLV Types

<table>
<thead>
<tr>
<th>Type</th>
<th>Name</th>
<th>Usage Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>reserved</td>
<td>-</td>
</tr>
<tr>
<td>1</td>
<td>BAS Function</td>
<td>Carries the BNG access functions to be enabled or disabled on specified interfaces.</td>
</tr>
<tr>
<td>2</td>
<td>Basic Subscriber</td>
<td>Carries the basic information about a BNG subscriber.</td>
</tr>
<tr>
<td>3</td>
<td>PPP Subscriber</td>
<td>Carries the PPP information</td>
</tr>
<tr>
<td>Code</td>
<td>Description</td>
<td>Details</td>
</tr>
<tr>
<td>------</td>
<td>--------------------------------------------------</td>
<td>----------------------------------------------</td>
</tr>
<tr>
<td>4</td>
<td>IPv4 Subscriber</td>
<td>Carries the IPv4 address of a BNG subscriber.</td>
</tr>
<tr>
<td>5</td>
<td>IPv6 Subscriber</td>
<td>Carries the IPv6 address of a BNG subscriber.</td>
</tr>
<tr>
<td>6</td>
<td>Subscriber Policy</td>
<td>Carries the policy information applied to a BNG subscriber.</td>
</tr>
<tr>
<td>7</td>
<td>IPv4 Routing</td>
<td>Carries the IPv4 network routing information.</td>
</tr>
<tr>
<td>8</td>
<td>IPv6 Routing</td>
<td>Carries the IPv6 network routing information.</td>
</tr>
<tr>
<td>9</td>
<td>IPv4 Static Subscriber Detect</td>
<td>Carries the IPv4 static subscriber detect information.</td>
</tr>
<tr>
<td>10</td>
<td>IPv6 Static Subscriber Detect</td>
<td>Carries the IPv6 static subscriber detect information.</td>
</tr>
<tr>
<td>11</td>
<td>L2TP-LAC Subscriber</td>
<td>Carries the L2TP LAC subscriber information.</td>
</tr>
<tr>
<td>12</td>
<td>L2TP-LNS Subscriber</td>
<td>Carries the L2TP LNS subscriber information.</td>
</tr>
<tr>
<td>13</td>
<td>L2TP-LAC-Tunnel</td>
<td>Carries the L2TP LAC tunnel subscriber information.</td>
</tr>
<tr>
<td>14</td>
<td>L2TP-LNS-Tunnel</td>
<td>Carries the L2TP LNS tunnel subscriber information.</td>
</tr>
<tr>
<td>15</td>
<td>Subscriber CGN Port Range</td>
<td>Carries the public IPv4 address and related port range of a BNG subscriber.</td>
</tr>
<tr>
<td>16-99</td>
<td>unassigned</td>
<td>-</td>
</tr>
<tr>
<td>100</td>
<td>Hello</td>
<td>Used for version and Keepalive timers negotiation.</td>
</tr>
<tr>
<td>101</td>
<td>Error Information</td>
<td>Carried in the Ack of the control message. Carries the processing result, success, or error.</td>
</tr>
<tr>
<td>102</td>
<td>Keepalive</td>
<td>Carried in the Hello message for Keepalive timers negotiation.</td>
</tr>
<tr>
<td>103-199</td>
<td>unassigned</td>
<td>-</td>
</tr>
<tr>
<td>200</td>
<td>Interface Status</td>
<td>Interfaces status reported by the UP including physical interfaces, sub-interfaces, trunk interfaces, and tunnel interfaces.</td>
</tr>
<tr>
<td>201</td>
<td>Board Status</td>
<td>Board information reported by the UP including the board type and in-position status.</td>
</tr>
<tr>
<td>202-299</td>
<td>unassigned</td>
<td>-</td>
</tr>
<tr>
<td>300</td>
<td>Subscriber Traffic Statistics</td>
<td>User traffic statistics.</td>
</tr>
<tr>
<td>301</td>
<td>Subscriber Detection Results</td>
<td>User detection information.</td>
</tr>
<tr>
<td>302</td>
<td>Update Response</td>
<td>The processing result of a subscriber session update.</td>
</tr>
<tr>
<td>Code</td>
<td>Operation</td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>reserved</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Update</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Delete</td>
<td></td>
</tr>
<tr>
<td>3-15</td>
<td>unassigned</td>
<td></td>
</tr>
</tbody>
</table>

9.3. TLV Operation Codes

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

See Section 7.3.

<table>
<thead>
<tr>
<th>Type</th>
<th>Name</th>
<th>Section of this document</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>reserved</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>VRF Name</td>
<td>7.3.1.</td>
</tr>
<tr>
<td>2</td>
<td>Ingress-QoS-Profile</td>
<td>7.3.1.</td>
</tr>
<tr>
<td>3</td>
<td>Egress-QoS-Profile</td>
<td>7.3.1.</td>
</tr>
<tr>
<td>4</td>
<td>User-ACL-Policy</td>
<td>7.3.1.</td>
</tr>
<tr>
<td>5</td>
<td>Multicast-ProfileV4</td>
<td>7.3.1.</td>
</tr>
<tr>
<td>6</td>
<td>Multicast-ProfileV6</td>
<td>7.3.1.</td>
</tr>
<tr>
<td>7</td>
<td>Ingress-CAR</td>
<td>7.3.2.</td>
</tr>
<tr>
<td>8</td>
<td>Egress-CAR</td>
<td>7.3.3.</td>
</tr>
<tr>
<td>9</td>
<td>NAT-Instance</td>
<td>7.3.1.</td>
</tr>
<tr>
<td>10</td>
<td>Pool-Name</td>
<td>7.3.1.</td>
</tr>
<tr>
<td>11</td>
<td>If-Desc</td>
<td>7.3.4.</td>
</tr>
<tr>
<td>12</td>
<td>IPv6-Address List</td>
<td>7.3.5.</td>
</tr>
<tr>
<td>13</td>
<td>Vendor</td>
<td>7.3.6.</td>
</tr>
<tr>
<td>12-65534</td>
<td>unassigned</td>
<td></td>
</tr>
<tr>
<td>65535</td>
<td>reserved</td>
<td></td>
</tr>
</tbody>
</table>

9.5.  Error Codes

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Success</td>
<td>Success</td>
</tr>
<tr>
<td>1</td>
<td>Fail</td>
<td>Malformed message received.</td>
</tr>
<tr>
<td>2</td>
<td>TLV-Unknown</td>
<td>One or more of the TLVs was not understood.</td>
</tr>
<tr>
<td>3</td>
<td>TLV-Length</td>
<td>The TLV length is abnormal.</td>
</tr>
<tr>
<td>4-999</td>
<td>unassigned</td>
<td>Unassigned basic error codes.</td>
</tr>
<tr>
<td>1000</td>
<td>reserved</td>
<td></td>
</tr>
<tr>
<td>1001</td>
<td>Version-Mismatch</td>
<td>The version negotiation fails. Terminate. The subsequent service processes corresponding to the UP are suspended.</td>
</tr>
<tr>
<td>1002</td>
<td>Keepalive Error</td>
<td>The keepalive negotiation fails.</td>
</tr>
<tr>
<td>1003</td>
<td>Timer Expires</td>
<td>The establishment timer expired.</td>
</tr>
<tr>
<td>1004-1999</td>
<td>unassigned</td>
<td>Unassigned error codes for version negotiation.</td>
</tr>
<tr>
<td>2000</td>
<td>reserved</td>
<td></td>
</tr>
<tr>
<td>2001</td>
<td>Synch-NoReady</td>
<td>The data to be smoothed is not ready.</td>
</tr>
<tr>
<td>2002</td>
<td>Synch-Unsupport</td>
<td>The request for smooth data is not supported.</td>
</tr>
<tr>
<td>2003-2999</td>
<td>unassigned</td>
<td>Unassigned data synchronization error codes.</td>
</tr>
</tbody>
</table>
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

9.6. If-Type Values

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

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>reserved</td>
</tr>
<tr>
<td>1</td>
<td>Fast Ethernet (FE)</td>
</tr>
<tr>
<td>2</td>
<td>GE</td>
</tr>
<tr>
<td>3</td>
<td>10GE</td>
</tr>
<tr>
<td>4</td>
<td>100GE</td>
</tr>
<tr>
<td>5</td>
<td>Eth-Trunk</td>
</tr>
<tr>
<td>6</td>
<td>Tunnel</td>
</tr>
<tr>
<td>7</td>
<td>VE</td>
</tr>
<tr>
<td>8-254</td>
<td>unassigned</td>
</tr>
<tr>
<td>255</td>
<td>reserved</td>
</tr>
</tbody>
</table>
9.7. Access-Mode Values

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

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Layer 2 subscriber</td>
</tr>
<tr>
<td>1</td>
<td>Layer 3 subscriber</td>
</tr>
<tr>
<td>2</td>
<td>Layer 2 leased line</td>
</tr>
<tr>
<td>3</td>
<td>Layer 3 leased line</td>
</tr>
<tr>
<td>4-254</td>
<td>unassigned</td>
</tr>
<tr>
<td>255</td>
<td>reserved.</td>
</tr>
</tbody>
</table>

9.8. Access Method Bits

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

Auth-Method4

<table>
<thead>
<tr>
<th>Bit</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x01</td>
<td>PPPoE authentication</td>
</tr>
<tr>
<td>0x02</td>
<td>DOT1X authentication</td>
</tr>
<tr>
<td>0x04</td>
<td>Web authentication</td>
</tr>
<tr>
<td>0x08</td>
<td>Web fast authentication</td>
</tr>
<tr>
<td>0x10</td>
<td>Binding authentication</td>
</tr>
<tr>
<td>0x20</td>
<td>reserved</td>
</tr>
<tr>
<td>0x40</td>
<td>reserved</td>
</tr>
<tr>
<td>0x80</td>
<td>reserved</td>
</tr>
</tbody>
</table>

Auth-Method6

<table>
<thead>
<tr>
<th>Bit</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x01</td>
<td>PPPoE authentication</td>
</tr>
<tr>
<td>0x02</td>
<td>DOT1X authentication</td>
</tr>
<tr>
<td>0x04</td>
<td>Web authentication</td>
</tr>
<tr>
<td>0x08</td>
<td>Web fast authentication</td>
</tr>
<tr>
<td>0x10</td>
<td>Binding authentication</td>
</tr>
<tr>
<td>0x20</td>
<td>reserved</td>
</tr>
<tr>
<td>0x40</td>
<td>reserved</td>
</tr>
<tr>
<td>0x80</td>
<td>reserved</td>
</tr>
</tbody>
</table>
### 9.9. Route-Type Values

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

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>User host route</td>
</tr>
<tr>
<td>1</td>
<td>Radius authorization FrameRoute</td>
</tr>
<tr>
<td>2</td>
<td>Network segment route</td>
</tr>
<tr>
<td>3</td>
<td>Gateway route</td>
</tr>
<tr>
<td>4</td>
<td>Radius authorized IP route</td>
</tr>
<tr>
<td>5</td>
<td>L2TP LNS side user route</td>
</tr>
<tr>
<td>6-65534</td>
<td>unassigned</td>
</tr>
<tr>
<td>65535</td>
<td>reserved</td>
</tr>
</tbody>
</table>

### 9.10. Access-Type Values

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

<table>
<thead>
<tr>
<th>Access-Type Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>reserved</td>
</tr>
<tr>
<td>1</td>
<td>PPP access (PPP [RFC1661])</td>
</tr>
<tr>
<td>2</td>
<td>PPP over Ethernet over ATM access (PPPoEoA)</td>
</tr>
<tr>
<td>3</td>
<td>PPP over ATM access (PPPoA [RFC3336])</td>
</tr>
<tr>
<td>4</td>
<td>PPP over Ethernet access (PPPoE [RFC2516])</td>
</tr>
<tr>
<td>5</td>
<td>PPPoE over VLAN access (PPPoEoVLAN [RFC2516])</td>
</tr>
<tr>
<td>6</td>
<td>PPP over LNS access (PPPoLNS)</td>
</tr>
<tr>
<td>7</td>
<td>IP over Ethernet DHCP access (IPoE_DHCP)</td>
</tr>
<tr>
<td>8</td>
<td>IP over Ethernet EAP authentication access (IPoE_EAP)</td>
</tr>
<tr>
<td>9</td>
<td>IP over Ethernet Layer 3 access (IPoE_L3)</td>
</tr>
<tr>
<td>10</td>
<td>IP over Ethernet Layer 2 Static access (IPoE_L2_STATIC)</td>
</tr>
<tr>
<td>11</td>
<td>Layer 2 Leased Line access (L2_Leased_Line)</td>
</tr>
<tr>
<td>12</td>
<td>Layer 2 VPN Leased Line access (L2VPN_Leased_Line)</td>
</tr>
<tr>
<td>13</td>
<td>Layer 3 Leased Line access (L3_Leased_Line)</td>
</tr>
<tr>
<td>14</td>
<td>Layer 2 Leased line Sub-User access (L2_Leased_Line_SUB_USER)</td>
</tr>
<tr>
<td>15</td>
<td>L2TP LAC tunnel access (L2TP_LAC)</td>
</tr>
<tr>
<td>16</td>
<td>L2TP LNS tunnel access (L2TP_LNS)</td>
</tr>
<tr>
<td>17-254</td>
<td>unassigned</td>
</tr>
<tr>
<td>255</td>
<td>reserved</td>
</tr>
</tbody>
</table>
10. IANA Considerations

This document requires no IANA actions.
11. 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

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

Email: lizhenqiang@chinamobile.com

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

Email: mach.chen@huawei.com

Zhouyi Yu
Huawei Technologies

Email: yuzhouyi@huawei.com

Chengguang Niu
Huawei Technologies

Email: chengguang.niu@huawei.com

Zitao Wang
Huawei Technologies

Email: wangzitao@huawei.com

Jun Song
Huawei Technologies

Email: song.jun@huawei.com

Dan Meng
H3C Technologies
No.1 Lixing Center
No.8 guangxun south street, Chaoyang District,
Beijing, 100102
China
Email: mengdan@h3c.com

Hanlei Liu
H3C Technologies
No.1 Lixing Center
No.8 guangxun south street, Chaoyang District,
Beijing, 100102
China

Email: hanlei_liu@h3c.com

Victor Lopez
Telefonica
Spain

Email: victor.lopezalvarez@telefonica.com
Acknowledgements

The helpful comments and suggestions of the following are hereby acknowledged:

Loa Andersson

Greg Mirsky
Normative References


Informative References


Hu, et al


Authors’ Addresses

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

Email: hushujun@chinamobile.com

Donald Eastlake, 3rd
Futurewei Technologies
2386 Panoramic Circle
Apopka, FL  32703
USA

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

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

Email: qinfengwei@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
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 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.