

Network Working Group  
Internet-Draft  
Intended status: Informational  
Expires: 26 August 2022

Y. Luo  
L. Qu  
China Telcom Co., Ltd.  
X. Huang  
Tencent  
G. Mishra  
Verizon Inc.  
H. Chen  
Futurewei  
S. Zhuang  
Z. Li  
Huawei  
22 February 2022

Architecture for Use of BGP as Central Controller  
draft-cth-rtgwg-bgp-control-08

Abstract

BGP is a core part of a network including Software-Defined Networking (SDN) system. It has the traffic engineering information on the network topology and can compute optimal paths for a given traffic flow across the network.

This document describes some reference architectures for BGP as a central controller. A BGP-based central controller can simplify the operations on the network and use network resources efficiently for providing services with high quality.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

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."

This Internet-Draft will expire on 26 August 2022.

#### Copyright Notice

Copyright (c) 2022 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 (<https://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 Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

#### Table of Contents

|  |    |
|--|----|
| 1. Introduction . . . . .                                    | 3  |
| 2. Terminology . . . . .                                     | 4  |
| 3. Architectures . . . . .                                   | 5  |
| 3.1. Building Blocks . . . . .                               | 5  |
| 3.1.1. TEDB . . . . .  | 5  |
| 3.1.2. SLDB . . . . .  | 5  |
| 3.1.3. TPDB . . . . .  | 5  |
| 3.1.4. CSPF . . . . .  | 6  |
| 3.1.5. TM . . . . .  | 6  |
| 3.2. One Controller . . . . .                                | 7  |
| 3.3. Controller Cluster . . . . .                            | 8  |
| 3.4. Hierarchical Controllers . . . . .                      | 11 |
| 4. Application Scenarios . . . . .                           | 12 |
| 4.1. Business-oriented Traffic Steering . . . . .            | 12 |
| 4.1.1. Preferential Users . . . . .                          | 12 |
| 4.1.2. Preferential Services . . . . .                       | 13 |
| 4.2. Traffic Congestion Mitigation . . . . .                 | 14 |
| 4.2.1. Congestion Mitigation in Core . . . . .               | 15 |
| 4.2.2. Congestion Mitigation among ISPs . . . . .            | 15 |
| 4.2.3. Congestion Mitigation at International Edge . . . . . | 16 |
| 5. Security Considerations . . . . .                         | 17 |
| 6. IANA Considerations . . . . .                             | 17 |
| 7. Acknowledgements . . . . .                                | 17 |
| 8. Contributors . . . . .                                    | 17 |
| 9. References . . . . .                                      | 17 |

|                                       |    |
|---------------------------------------|----|
| 9.1. Normative References . . . . .   | 17 |
| 9.2. Informative References . . . . . | 18 |
| Authors' Addresses . . . . .          | 19 |

## 1. Introduction

Border Gateway Protocol (BGP) [RFC1771] is an exterior gateway protocol (EGP). It is developed to exchange routing information among routers in different autonomous systems (ASes). Along its developments, BGP has been extended to provide numerous new functions. It collects the link states including traffic engineering (TE) information from other protocols such as IGP and distributes them among routers in different ASes [RFC7752]. It also controls the redirection of traffic flows [RFC5575]. Furthermore, it distributes MPLS labels [RFC3107]. For scalability, BGP is extended to have Route Reflector (RR) [RFC4456].

For segment routing (SR), BGP is extended to advertise SR policies with candidate paths to the policy headend routers, which are typically ingress routers [I-D.ietf-idr-segment-routing-te-policy]. The SR specific PCEP extensions are defined in [I-D.ietf-pce-segment-routing]. A stateful PCE can compute an SR traffic engineering (SR-TE) path satisfying a set of constraints, and initiate an SR-TE path on a headend router using the extensions.

An SDN controller (or controller for short) is the core of an SDN system or network. It is between network elements (NEs) such as routers or switches at one end and applications such as Operational Support System (OSS) or Network Management System (NMS) at the other end. The essential function of a controller is to steer traffic flows across the network for providing more services with higher quality. It manages network resources such as link bandwidth, computes expected paths for carrying traffic flows based on available network resources, programs the network elements for the creation of tunnels along the paths, and redirects traffic flows into corresponding tunnels.

Based on the current BGP, it is natural, beneficial and relatively simple to extend BGP to become a controller. Using BGP as a controller for a network will greatly simplify the operations on the network. It avoids deploying, operating and maintaining a new extra component or protocol such as PCE as a controller in the network.

This document describes some reference architectures for BGP as a central controller and introduces some scenarios to which the BGP controller can be applied.

## 2. Terminology

- \* SR: Segment Routing
- \* RR: Route Reflector
- \* SID: Segment Identifier
- \* SR-Path: Segment Routing Path
- \* SR-Tunnel: Segment Routing Tunnel
- \* TEDB: Traffic Engineering Database
- \* LSDB: Link State Database
- \* SLDB: SID/Label Database
- \* TPDB: Tunnel and Path Database
- \* CSPF: Constrained Shortest Path First
- \* TM: Tunnel Manager
- \* NMS: Network Management System
- \* SRLB: SR Local Block
- \* NE: Network Element
- \* PCE: Path Computation Element
- \* AS: Autonomous System
- \* QoS: Quality of Service
- \* ISP: Internet Service Provider
- \* MAN: Metropolitan Area Network
- \* OTT: Over the Top
- \* OTTSP: Over the Top Service Provider, or Content Operator
- \* AR: Access Router

### 3. Architectures

An architecture for the use of BGP as a central controller is based on the essential function of a controller. It is constructed from some building blocks or components. After introduction to building blocks, a few of reference architectures are described in this section.

#### 3.1. Building Blocks

Some critical building blocks are briefed. They are Traffic Engineering Database (TEDB or TED for short), SID/Label Database (SLDB), Tunnel and Path Database (TPDB), Constrained Shortest Path First (CSPF), and Tunnel Manager (TM).

##### 3.1.1. TEDB

The Traffic Engineering Database (TEDB) stores the Traffic Engineering (TE) information about the network. It includes the unreserved bandwidth at each of eight priority levels for every link in the network.

TEDB can be an individual block, which is constructed from the link state information received. It may be embedded into the link state database (LSDB) in the BGP when the BGP creates/updates the LSDB from the link state information it receives.

##### 3.1.2. SLDB

The SID/Label Database (SLDB) records and maintains the status of every Segment Identifier (SID) and label for every node, interface/link and/or prefix in the network, which the controller controls. The status of SID/label indicates whether the SID/Label is assigned. If it is assigned, then the object such as the node, link or prefix, to which it is assigned, is recorded.

SLDB can be an individual block, which is constructed from the link state information such as SR Local Block (SRLB) that the BGP receives. It may be embedded into the link state database (LSDB) in the BGP when the BGP creates the LSDB from the link state information it receives.

##### 3.1.3. TPDB

The Tunnel and Path Database (TPDB) stores the information for every tunnel, which includes:

- o the parameters received for the tunnel from a user/application,

- o the path computed for the tunnel,
- o the resources such as link bandwidth reserved along the path for the tunnel,
- o the SID/labels assigned along the path for the tunnel, and
- o the status of the tunnel.

#### 3.1.4. CSPF

The Constrained Shortest Path First (CSPF) computes a path for a tunnel such as SR tunnel or LSP tunnel that satisfies a set of given constraints using the information in TEDB.

#### 3.1.5. TM

The Tunnel Manager (TM) receives a request for an operation on a tunnel from a user or an application such as Network Management System (NMS). The operation may be a creation of a new tunnel, a deletion of an existing tunnel, or a change to an existing tunnel.

When receiving a request for creating a new tunnel, the TM asks the CSPF to compute a path for the tunnel that satisfies the constraints given for the tunnel.

After obtaining the path for the tunnel from the CSPF, the TM requests the SLDB to assign SID/labels along the path for the tunnel and asks the TEDB to reserve the resources such as link bandwidth along the path for the tunnel.

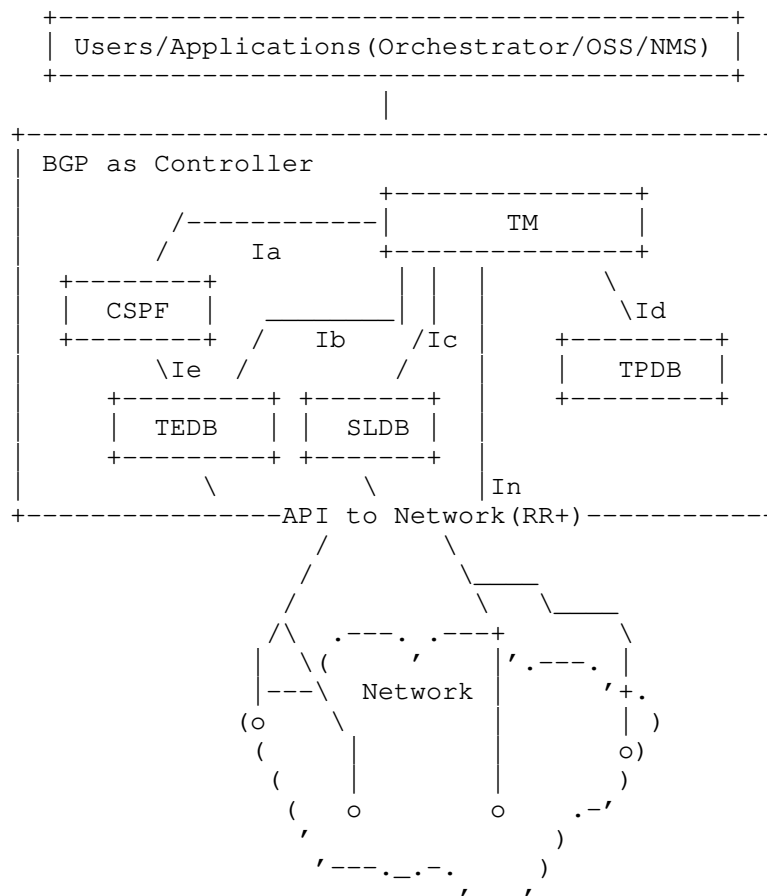
The TM in a central controller may set up the tunnel along the path in the network by programming each of the NEs along the path through the API to the network. In a SR network, the TM initiates a SR tunnel in the network by sending a sequence of SID/labels to the source NE of the tunnel.

The TM records the information for the tunnel in the Tunnel and Path Database (TPDB). The information includes the path computed for the tunnel, the resources such as bandwidth reserved along the path, the SID/labels assigned along the path for the tunnel, and the status of the tunnel.

### 3.2. One Controller

Figure below illustrates a reference architecture for using the BGP as a central controller, which controls a network. The BGP as a controller in the reference architecture controls a network through an API to the network such as BGP+/RR+ (extensions to BGP for central controller). The BGP controller is responsible for creating and maintaining every tunnel in the network. It also controls the redirection of traffic flow to each tunnel.

The BGP controller comprises a number of modules, including a TM, a CSPF, a TEDB, a SLDB and a TPDB. The interfaces among these modules are listed as follows:



- \* Interface Ia between the TM and the CSPF. Through this interface, the TM requests the CSPF to compute a path for a tunnel with a set of constraints, and the CSPF responses the TM with the path computed that satisfies the constraints.
- \* Interface Ib between the TM and the TEDB. When a tunnel is to be created, through this interface, the TM reserves in the TEDB the TE resources such as link bandwidths on every link along the path computed for the tunnel. When a tunnel is deleted, the TM releases the TE resources such as link bandwidths on every link along the path for the tunnel.
- \* Interface Ic between the TM and the SLDB. When a tunnel is to be created, through this interface, the TM reserves in the SLDB a SID/label for every link or some links along the path computed for the tunnel. When a tunnel is deleted, the TM releases the SID/label for every link or some links along the path for the tunnel.
- \* Interface Id between the TM and the TPDB. the TM updates the information for every tunnel in the TPDB through this interface.
- \* Interface Ie between the CSPF and the TEDB. Through this interface, the CSPF accesses the traffic engineering information such as link bandwidths when it computes a path for a tunnel.

There is an interface In between the BGP controller and the network. In fact, there is a control channel (or interface) between the BGP controller and every (edge) node in the network.

Initially, the TEDB obtains the original traffic engineering (TE) information such as link bandwidths from the network through the interface In (i.e., API to network) for every link in the network. The SLDB gets the original SID/label resources from the network through the interface for every node, link and prefix in the network.

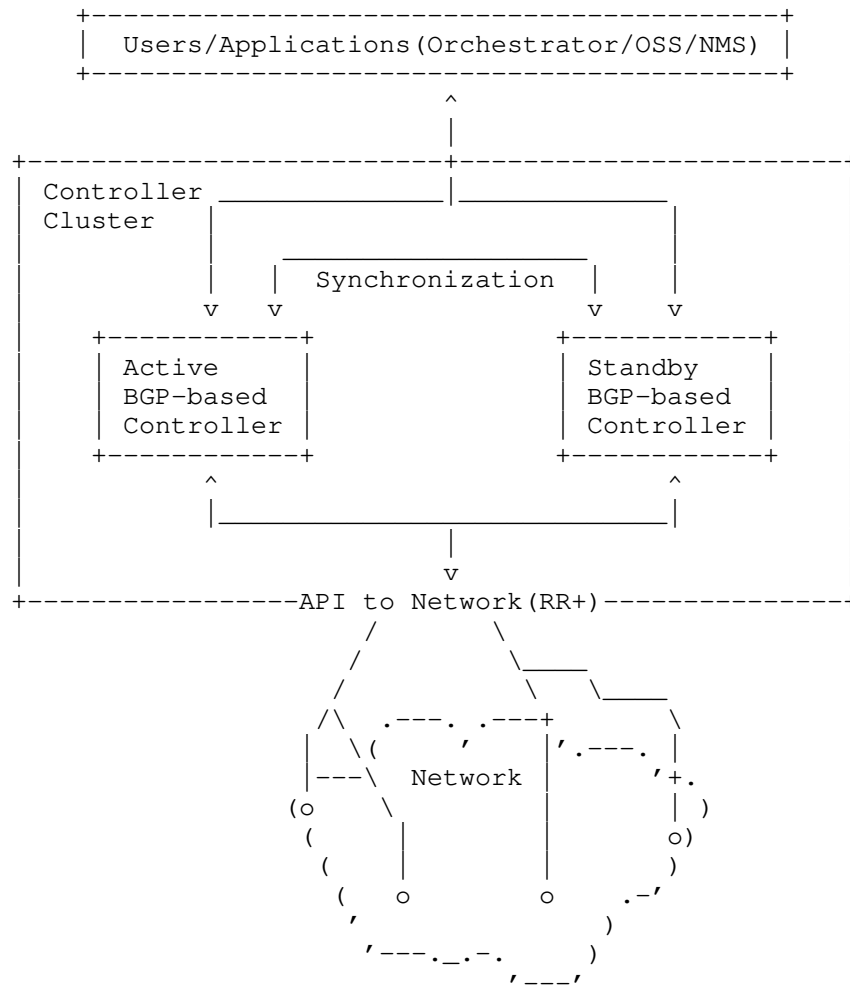
### 3.3. Controller Cluster

A critical issue in a network with a central controller is the failure of the controller, which is a single point of failure (SPOF). If the controller fails, the entire network may not work.



A controller cluster (i.e., a group of controllers) works as a single controller from user's point of view. A simple controller cluster consists of two controllers. One works as a active (or say primary) controller, and the other as a standby (or say secondary) controller. In normal operations, the active controller is responsible for the network it controls. It also synchronizes with the standby controller. When the active controller fails, the standby controller becomes a new active controller, which controls the network.

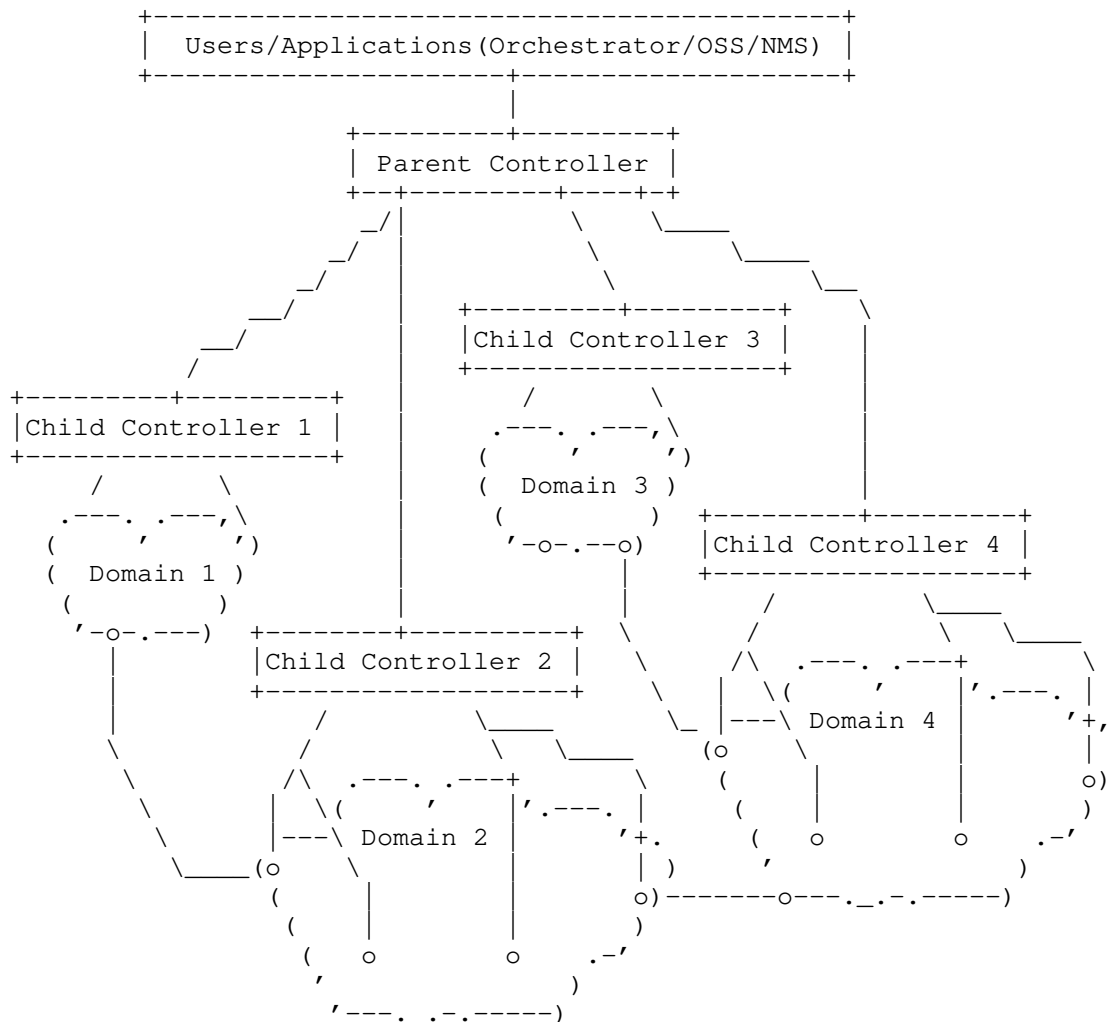
The Figure below illustrates a simple controller cluster containing two BGP-based controllers: Active BGP-based Controller and Standby BGP-based Controller. In normal operations, the active controller interacts with users and/or applications. For example, it receives configurations for tunnels and the traffic flows to tunnels from users. The active controller instructs the network elements in the network to provide the services requested by users and/or applications. For example, after receiving the configurations for a tunnel and a traffic flow to the tunnel, the active controller computes a path for the tunnel, programs (or say instructs) the network elements along the path for creating the tunnel, and instructs the ingress of the tunnel to direct the traffic flow into the tunnel.



During this process, the status information about the network is updated in the active controller. The information includes: the traffic engineering information in their TEDBs, the SID/label information in their SLDBs, and the configurations, paths, resources and status for tunnels in their TPDBs. The active controller synchronizes this information with the standby controller. Thus these two controllers have the same status information about the network. When the active controller fails, the standby controller takes over the role of the active controller smoothly and becomes active controller.

### 3.4. Hierarchical Controllers

The Figure below illustrates a system with hierarchical controllers. There is one Parent Controller and four Child Controllers: Child Controller 1, Child Controller 2, Child Controller 3 and Child Controller 4.



The parent controller communicates with these four child controllers and controls them, each of which controls (or is responsible for) a domain. Child controller 1 controls domain 1, Child controller 2 controls domain 2, Child controller 3 controls domain 3, and Child controller 4 controls domain 4.

One level of hierarchy of controllers is illustrated in the figure above. There is one parent controller at top level, which is not a child controller. Under the parent controller, there are four child controllers, which are not parent controllers.

In a general case, at top level there is one parent controller that is not a child controller, there are some controllers that are both parent controllers and child controllers, and there are a number of child controllers that are not parent controllers. This is a system of multiple levels of hierarchies, in which one parent controller controls or communicates with a first number of child controllers, some of which are also parent controllers, each of which controls or communicates with a second number of child controllers, and so on.

The parent controller receives requests for creating end to end tunnels from users or applications. For each request, the parent controller is responsible for obtaining a path for the tunnel and creating the tunnel along the path through sending instructions to the corresponding child controllers.

#### 4. Application Scenarios

This section introduces a set of scenarios to which the controller can be applied.

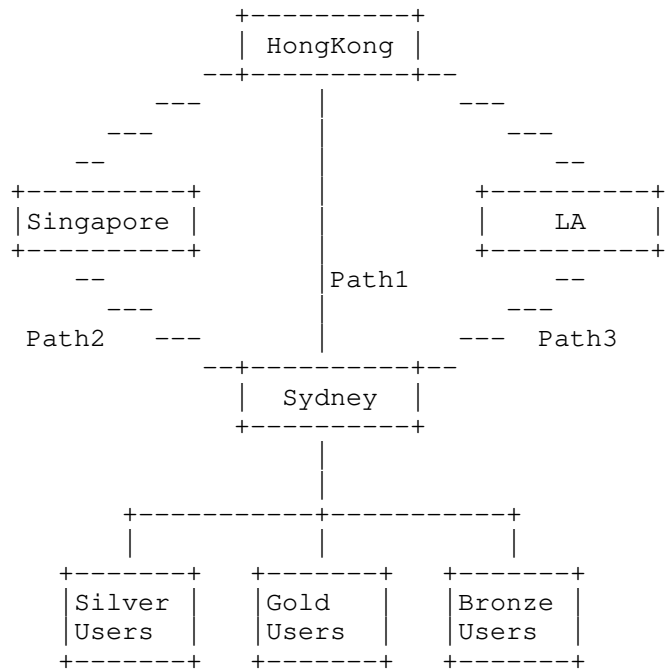
##### 4.1. Business-oriented Traffic Steering

It is reasonable in commercial sense to provide multiple paths to the same destination with differentiated experiences for preferential users/services. This is an efficient approach to maximize providers' network resource usage as well as their profit and offer more choices to network users.

###### 4.1.1. Preferential Users

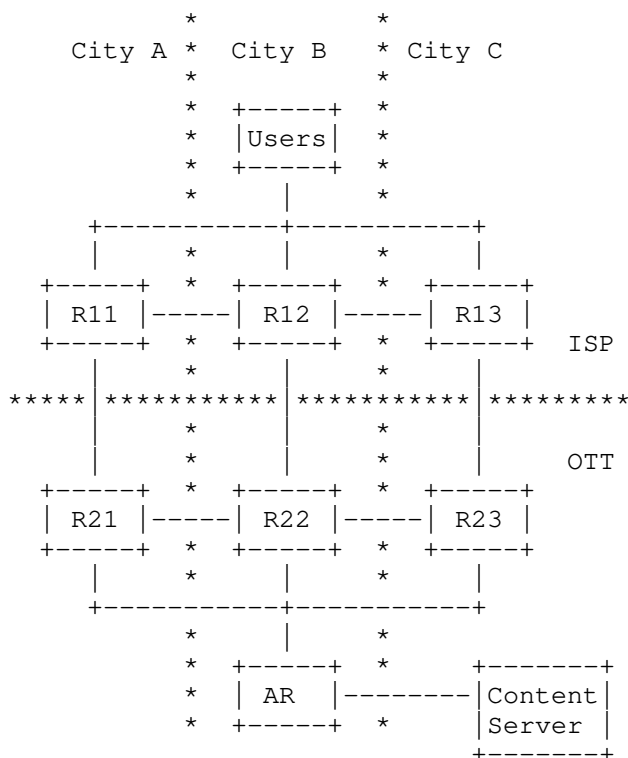
In the Figure below for an ISP network, there are three kinds of users in Sydney, saying Gold, Silver and Bronze, and they wish to visit website located in HongKong. The ISP provides three different paths with different experiences according to users' priority. The Gold Users may use Path1 with less latency and loss. The Silver Users may use the Path2 through Singapore with less latency but maybe some congestion there. The Bronze Users may use Path3 through LA

with some latency and loss.



4.1.2. Preferential Services

As depicted in the Figure below, the OTTSP has 3 exits with one ISP, which are located in City A, City B and City C. The content is obtained from Content Server and send to the exits through AR. An OTTSP may make its steering strategy based on different services. For example, the OTTSP in the Figure may choose exit R21 for video service and exit R22 for web service, which REQUIRES a mechanism/ system exists to identify different services from traffic flow.



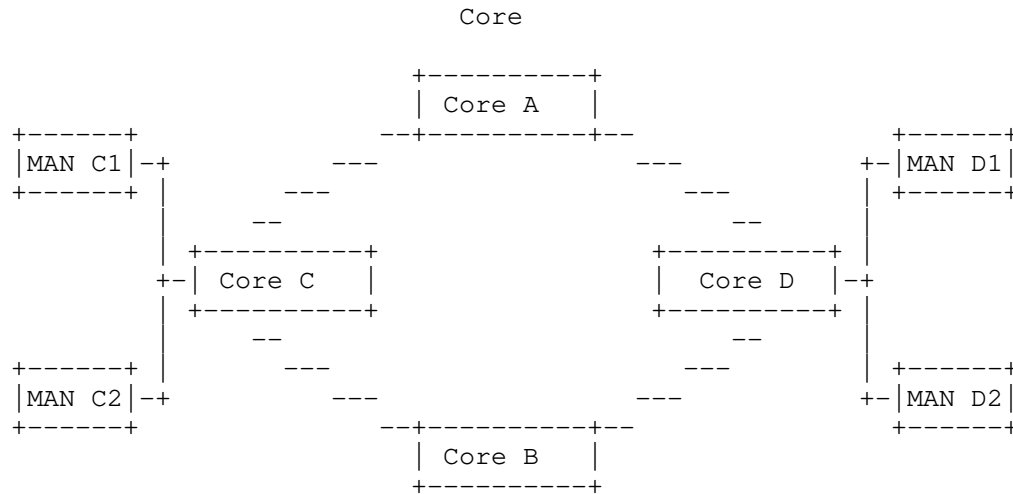
#### 4.2. Traffic Congestion Mitigation

It is a persistent goal for providers to increase the utilization ratio of their current network resources, and to mitigate the traffic congestion. Traffic congestion is possible to happen anywhere in the ISP network (MAN, IDC, core and the links between them), because internet traffic is hard to predict. For example, there might be some local online events that the network operators didn't know beforehand, or some sudden attack just happened. Even for the big events that can be predicted, such as annual online discount of e-commerce company, or IOS update of Apple Inc, we could not guarantee there is no congestion. Since the network capacity expansion is usually an annual operation, there could be delay on any links of the engineering. As a result, the temporary traffic steering is always needed. The same thing happens to the OTT networks as well.

It should be noted that, the traffic steering is absolutely not a global behavior. It just acts on part of the network, and it's temporary.

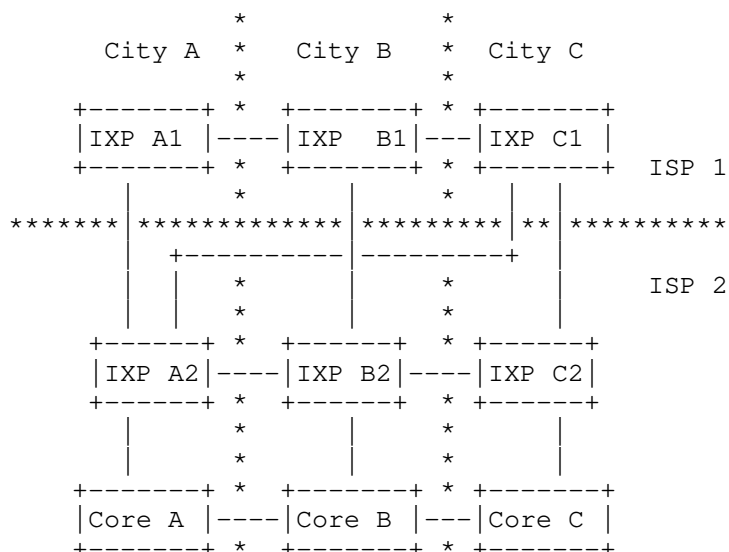
#### 4.2.1. Congestion Mitigation in Core

As depicted in the Figure below, traffic from MAN C1 to MAN D2 follows the path Core C->Core B->Core D as the primary path, but somehow the load ratio becomes too much. It is reasonable to transfer some traffic load to less utilized path Core C->Core A->Core D when the primary path has congestion.



#### 4.2.2. Congestion Mitigation among ISPs

As depicted in the Figure below, ISP1 and ISP2 are interconnect by 3 exits which are located in 3 cities respectively. The links between ISP1 and ISP2 in the same city are called local links, and the rest are long distance links. Traffic from IXP C1 to Core A in ISP 2 usually passes through link IXP C1->IXP A2->Core A. This is a long distant route, directly connecting city C and city A. Part of traffic could be transferred to link IXP.

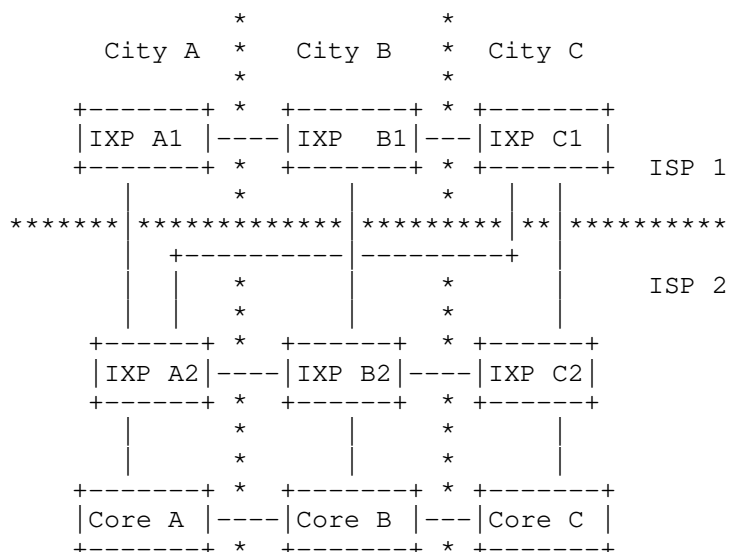


#### 4.2.3. Congestion Mitigation at International Edge

An ISP usually interconnects with more than 2 transit networks at the international edge, so it is quite common that multiple paths may exist for the same foreign destination. Usually those paths with better QoS properties such as latency, loss, jitter and etc are often preferred. Since these properties keep changing from time to time, the decision of path selection has to be made dynamically.

As depicted in the Figure below, the traffic to the foreign destination H from IP core network (AS C1) has two choices on transit network, saying Transit A and Transit B. Under normal conditions, Transit B is the primary choice, but Transit A will be preferred when the QoS of Transit B gets worse. As a result, the same traffic will go through Transit A instead.





## 5. Security Considerations

The interactions with a BGP-based controller are similar to those with any other SDN controller. The security implications of SDN controller have not been fully discussed or described. Therefore, protocol and applicability for solutions around this architecture must take proper account of these concerns.

## 6. IANA Considerations

This document does not require any IANA actions.

## 7. Acknowledgements

The authors would like to thank Chris Bowers, Jeff Tantsura for their valuable suggestions and comments on this draft.

## 8. Contributors

Nan Wu  
Huawei  
Email: eric.wu@huawei.com

## 9. References

### 9.1. Normative References

- [RFC1771] Rekhter, Y. and T. Li, "A Border Gateway Protocol 4 (BGP-4)", RFC 1771, DOI 10.17487/RFC1771, March 1995, <<https://www.rfc-editor.org/info/rfc1771>>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC3107] Rekhter, Y. and E. Rosen, "Carrying Label Information in BGP-4", RFC 3107, DOI 10.17487/RFC3107, May 2001, <<https://www.rfc-editor.org/info/rfc3107>>.
- [RFC4271] Rekhter, Y., Ed., Li, T., Ed., and S. Hares, Ed., "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, DOI 10.17487/RFC4271, January 2006, <<https://www.rfc-editor.org/info/rfc4271>>.
- [RFC4456] Bates, T., Chen, E., and R. Chandra, "BGP Route Reflection: An Alternative to Full Mesh Internal BGP (IBGP)", RFC 4456, DOI 10.17487/RFC4456, April 2006, <<https://www.rfc-editor.org/info/rfc4456>>.
- [RFC5575] Marques, P., Sheth, N., Raszuk, R., Greene, B., Mauch, J., and D. McPherson, "Dissemination of Flow Specification Rules", RFC 5575, DOI 10.17487/RFC5575, August 2009, <<https://www.rfc-editor.org/info/rfc5575>>.
- [RFC7752] Gredler, H., Ed., Medved, J., Previdi, S., Farrel, A., and S. Ray, "North-Bound Distribution of Link-State and Traffic Engineering (TE) Information Using BGP", RFC 7752, DOI 10.17487/RFC7752, March 2016, <<https://www.rfc-editor.org/info/rfc7752>>.

## 9.2. Informative References

- [I-D.ietf-idr-bgppls-segment-routing-epe]  
Previdi, S., Talaulikar, K., Filsfils, C., Patel, K., Ray, S., and J. Dong, "Border Gateway Protocol - Link State (BGP-LS) Extensions for Segment Routing BGP Egress Peer Engineering", Work in Progress, Internet-Draft, draft-ietf-idr-bgppls-segment-routing-epe-19, 16 May 2019, <<https://www.ietf.org/archive/id/draft-ietf-idr-bgppls-segment-routing-epe-19.txt>>.
- [I-D.ietf-idr-flowspec-path-redirect]  
Velde, G. V. D., Patel, K., and Z. Li, "Flowspec Indirection-id Redirect", Work in Progress, Internet-

Draft, draft-ietf-idr-flowspec-path-redirect-11, 26 May 2020, <<https://www.ietf.org/archive/id/draft-ietf-idr-flowspec-path-redirect-11.txt>>.

[I-D.ietf-idr-segment-routing-te-policy]

Previdi, S., Filsfils, C., Talaulikar, K., Mattes, P., Jain, D., and S. Lin, "Advertising Segment Routing Policies in BGP", Work in Progress, Internet-Draft, draft-ietf-idr-segment-routing-te-policy-14, 10 November 2021, <<https://www.ietf.org/archive/id/draft-ietf-idr-segment-routing-te-policy-14.txt>>.

[I-D.ietf-isis-segment-routing-extensions]

Previdi, S., Ginsberg, L., Filsfils, C., Bashandy, A., Gredler, H., and B. Decraene, "IS-IS Extensions for Segment Routing", Work in Progress, Internet-Draft, draft-ietf-isis-segment-routing-extensions-25, 19 May 2019, <<https://www.ietf.org/archive/id/draft-ietf-isis-segment-routing-extensions-25.txt>>.

[I-D.ietf-pce-segment-routing]

Sivabalan, S., Filsfils, C., Tantsura, J., Henderickx, W., and J. Hardwick, "Path Computation Element Communication Protocol (PCEP) Extensions for Segment Routing", Work in Progress, Internet-Draft, draft-ietf-pce-segment-routing-16, 4 March 2019, <<https://www.ietf.org/archive/id/draft-ietf-pce-segment-routing-16.txt>>.

[I-D.ietf-rtgwg-bgp-routing-large-dc]

Lapukhov, P., Premji, A., and J. Mitchell, "Use of BGP for Routing in Large-Scale Data Centers", Work in Progress, Internet-Draft, draft-ietf-rtgwg-bgp-routing-large-dc-11, 4 June 2016, <<https://www.ietf.org/archive/id/draft-ietf-rtgwg-bgp-routing-large-dc-11.txt>>.

[I-D.ietf-spring-segment-routing]

Filsfils, C., Previdi, S., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", Work in Progress, Internet-Draft, draft-ietf-spring-segment-routing-15, 25 January 2018, <<https://www.ietf.org/archive/id/draft-ietf-spring-segment-routing-15.txt>>.

Authors' Addresses

Yujia  
China Telcom Co., Ltd.  
109 West Zhongshan Ave, Tianhe District

Guangzhou  
510630  
China  
Email: luoyuj@sdu.edu.cn

Liang  
China Telcom Co., Ltd.  
109 West Zhongshan Ave, Tianhe District  
Guangzhou  
510630  
China  
Email: ouliang@chinatelecom.cn

Xiang  
Tencent  
Email: terranhuang@tencent.com

Gyan S. Mishra  
Verizon Inc.  
13101 Columbia Pike  
Silver Spring, MD 20904  
United States of America  
Phone: 301 502-1347  
Email: gyan.s.mishra@verizon.com

Huaimo Chen  
Futurewei  
Boston, MA,  
United States of America  
Email: Huaimo.chen@futurewei.com

Shunwan Zhuang  
Huawei  
Huawei Bld., No.156 Beiqing Rd.  
Beijing  
100095  
China  
Email: zhuangshunwan@huawei.com

Zhenbin Li  
Huawei  
Huawei Bld., No.156 Beiqing Rd.

Beijing  
100095  
China  
Email: lizhenbin@huawei.com

Internet Engineering Task Force  
Internet-Draft  
Intended status: Informational  
Expires: May 7, 2020

E. Haleplidis  
  
J. Hadi Salim  
Mojatatu Networks  
J. Chung  
Viasat  
November 4, 2019

ForCES-based BNG  
draft-haleplidis-forces-bng-00

## Abstract

This document provides an approach for the separation of the forwarding and control plane for the Broadband Network Gateway (BNG) using IETF's ForCES architecture. The document provides an initial primer on ForCES as well as an initial ForCES XML model that describes some basic functions of the BNG.

## Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

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."

This Internet-Draft will expire on May 7, 2020.

## Copyright Notice

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 (<https://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.

## Table of Contents

|   |    |
|---|----|
| 1. Introduction . . . . .                           | 3  |
| 2. Terminology and Conventions . . . . .            | 3  |
| 2.1. Requirements Language . . . . .                | 3  |
| 2.2. Definitions . . . . .                          | 3  |
| 3. ForCES overview . . . . .                        | 4  |
| 3.1. ForCES protocol . . . . .                      | 5  |
| 3.2. ForCES Model . . . . .                         | 7  |
| 3.3. ForCES & BNG . . . . .                         | 8  |
| 4. Basic BNG ForCES model . . . . .                 | 9  |
| 4.1. Authentication . . . . .                       | 11 |
| 4.1.1. PPPoE Discovery stage . . . . .              | 12 |
| 4.1.2. PPP Session stage . . . . .                  | 13 |
| 4.2. Subscriber Information Configuration . . . . . | 13 |
| 4.2.1. Supporting multiple access types . . . . .   | 16 |
| 4.3. Traffic monitoring . . . . .                   | 17 |
| 5. Advanced BNG Services . . . . .                  | 17 |
| 5.1. Bandwidth Management Service . . . . .         | 17 |
| 5.2. Stateless access control service . . . . .     | 19 |
| 5.3. Quota Enforcement service . . . . .            | 19 |
| 5.4. Lawful Intercept service . . . . .             | 20 |
| 6. LFB Class Descriptions . . . . .                 | 20 |
| 6.1. Port LFB . . . . .                             | 20 |
| 6.1.1. Data Handling . . . . .                      | 20 |
| 6.1.2. Components . . . . .                         | 21 |
| 6.1.3. Capabilities . . . . .                       | 22 |
| 6.1.4. Events . . . . .                             | 22 |
| 6.2. Classifier LFB . . . . .                       | 22 |
| 6.2.1. Data Handling . . . . .                      | 23 |
| 6.2.2. Components . . . . .                         | 23 |
| 6.2.3. Capabilities . . . . .                       | 24 |
| 6.2.4. Events . . . . .                             | 24 |
| 6.3. PPPoE LFB . . . . .                            | 24 |
| 6.3.1. Data Handling . . . . .                      | 24 |
| 6.3.2. Components . . . . .                         | 25 |
| 6.3.3. Capabilities . . . . .                       | 26 |
| 6.3.4. Events . . . . .                             | 26 |
| 6.4. IPv4 Routing LFB . . . . .                     | 26 |
| 6.4.1. Data Handling . . . . .                      | 27 |
| 6.4.2. Components . . . . .                         | 27 |
| 6.4.3. Capabilities . . . . .                       | 28 |
| 6.4.4. Events . . . . .                             | 28 |
| 6.5. Policer LFB . . . . .                          | 28 |

|  |    |
|--|----|
| 6.5.1. Data Handling . . . . .         | 28 |
| 6.5.2. Components . . . . .            | 28 |
| 6.5.3. Capabilities . . . . .          | 29 |
| 6.5.4. Events . . . . .                | 29 |
| 7. BNG ForCES XML model . . . . .      | 29 |
| 8. Acknowledgements . . . . .          | 56 |
| 9. IANA Considerations . . . . .       | 56 |
| 10. Security Considerations . . . . .  | 56 |
| 11. References . . . . .               | 56 |
| 11.1. Normative References . . . . .   | 56 |
| 11.2. Informative References . . . . . | 57 |
| Authors' Addresses . . . . .           | 58 |

## 1. Introduction

This document presents IETF's ForCES architecture as a basis for the control and forwarding separation for the Disaggregated Broadband Network Gateway (BNG) [TR-101].

For contextual overview, any prescribed "experience" in this document are based on deployment experience over many years at large and small deployment environments for embedded, cloud as well as data centre environments. Some of these deployments (still operational at time of writing) have been publicly hinted at in media [media1], [media2] and [media3].

## 2. Terminology and Conventions

### 2.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

### 2.2. Definitions

This document reiterates the terminology defined by the ForCES architecture in various documents for the sake of clarity.

FE Model - The ForCES model used for describing resources to be managed/controlled. This includes three components; the modeling of individual Logical Functional Block (LFB model), the logical interconnection between LFBs (LFB topology), and the FE-level attributes, including LFB components, capabilities and events. The FE model provides the basis to define the information elements exchanged between CEs and FEs in the ForCES protocol [RFC5810].



LFB (Logical Functional Block) Class - A template that represents a resource that is being modeled. Most LFBs relate to packet processing in the data path; however, that is not always the case. LFB classes are the basic building blocks of the FE model.

LFB Instance - A runtime instantiation of an LFB class.

ForCES Component - A ForCES Component is a well-defined, uniquely identifiable and addressable ForCES model building block. A component has a 32-bit ID, name, type, and an optional synopsis description. These are often referred to simply as components.

ForCES Protocol - Protocol that runs in the Fp reference points in the ForCES Framework [RFC3746].

ForCES Protocol Layer (ForCES PL) - A layer in the ForCES protocol architecture that defines the ForCES protocol messages, the protocol state transfer scheme, and the ForCES protocol architecture itself as defined in the ForCES Protocol Specification [RFC5810].

ForCES Protocol Transport Mapping Layer (ForCES TML) - A layer in ForCES protocol architecture that uses the capabilities of existing transport protocols to specifically address protocol message transportation issues, such as how the protocol messages are mapped to different transport media (like TCP, IP, ATM, Ethernet, etc.), and how to achieve and implement reliability, ordering, etc. the ForCES SCTP TML [RFC5811] describes a TML that is mandated for ForCES.

Broadband Network Gateway (BNG) - is the network edge aggregation point used by subscribers as the access point through which they connect to the broadband network.

### 3. ForCES overview

In this section we present a quick overview of the ForCES architecture. The reader is encouraged to read the relevant documents, in particular [RFC5810], [RFC5812] and [RFC5811].

The origins of ForCES lie in the desire to separate control and datapath; where "datapath" was intended to be packet processing resources. Over time, however, due to the convenience of the ForCES architecture it has been used for controlling and managing arbitrary (other than packet processing) resources. As long as one can abstract the resources using the ForCES model, the protocol semantics allows using ForCES protocol to control and manage said resources.

In the case of the BNG, we will show later the attributes such as interfaces, user statistics and QoS parameters can all be modeled as resources.

The ForCES architecture is comprised of:

1. A data model definition [RFC5812] serving as a basis for the architecture constructs acted on by the protocol.
2. The ForCES protocol (PL) [RFC5810] which acts on the model component constructs for the purpose of control/management.
3. A transport mapping layer (TML) which takes the PL constructs and maps them to underlying convenient transport(s) and then delivers them to the target end points. Currently there is only one standardized TML based on SCTP; [RFC5811]. however more could be defined – as an example QUIC [I-D.ietf-quic-transport] appears to be a very good fit.

### 3.1. ForCES protocol

The ForCES protocol features can be summarized as:

1. Transport independence. The ForCES protocol is intended to run on a variety of chosen protocols.
2. Simplified ForCES layer when possible:
  - \* Security is left up to the transport choice keeping the ForCES layer simple.
  - \* Optional configurable Controller high availability. FEs(resource owners) when desired can connect to multiple controllers in both cold or hot standby mode [RFC5810], [RFC7121].
3. Degrees of reliability. Deployment experience with ForCES as depicted in the SCTP TML (RFC 5811 [RFC5811]) has shown an absolute need for a variety of shades of reliability.
4. Node overload. Deployment experience has shown the need to protect against node overload in a work-conserving mode (thus optimal resource usage).
5. Transactional capabilities in the form of 2 phase commits.

6. Wire Serialization and optimization. Encoding on the wire is binary. The data model is sufficient to describe the content on the wire.
7. Transactional scalability, low latency and high throughput.
8. Various execution modes for transactions {Execute-all-or-none, Execute-until-error, Execute-all-despite-errors}.
9. Communication methods. ForCES provides two communication methods for a controller to receive data from the device, namely request/response and publish/subscribe. ForCES allows the controller at any time to access (request) any resource, and allows for a controller to subscribe to any supported resources events.
10. Simple and powerful API. The ForCES architecture provides (very) few simple protocol verbs which act upon a multitude of nouns as represented by the ForCES model. The grammar could be described as:

<Command> <Resource path> [Data]

In other words, the ForCES semantic allows composing of rich services on top of the basic grammar. The expressive simplicity of the protocol is achieved due to the few verbs which act on the agreed-to modeled LFB components. The protocol is totally agnostic of the nature of the resource being controlled/managed. It is up to the modeler to describe the resource in the manner that is fitting (although frowned-upon, one could describe the resource model exactly as it is implemented and reduce the generalities and therefore translation overhead). The model is highly extensible and for this reason, the knowledge of the resource control is offloaded to the service layer and a basic infrastructure is all that is needed.

The ForCES verbs are: {GET, SET, DELETE, REPORT and a few helpers}.

11. Traffic sensitive protocol level connectivity detection. ForCES layer heartbeats could be sent in either or both directions. Heartbeats, however are only sent when the link is idle.
12. Dynamic association of FEs to CEs. FEs register and unregister to controllers and advertise their capabilities and capacities.

### 3.2. ForCES Model

The ForCES Model features can be summarized as:

1. Data model modularity through LFB class definition.
2. Data model type definitions via atomic types, complex/compound types, grouping of compound types in the form of structures and indexed/keyed tables which then end up composing addressable components within an LFB class.
3. Hierarchical/tree definition of control/config/state components which is acted on by a controller via the ForCES protocol.
4. Information-modeled metadata and expectations.
5. Built in LFB class resource capability definition/advertisement.
6. Publish/subscribe LFB event model with expressive trigger and report definitions. Filters include:
  - \* Hysteresis - used to suppress generation of notifications for oscillations around a condition value. Example, "generate an event if the link laser goes below 10 or goes above 15".
  - \* Count - used to suppress event generation around occurrence count. Example, "report the event to the client only after 5 consecutive occurrences".
  - \* Time interval - used to suppress event generation around a time limit. Example, "Generate an event if table foo row hasn't been used in the last 10 minutes".
7. Data model flexibility/extensibility through augmentations, and inheritance.
8. Backward and forward compatibility via LFB design and versioning rules.
9. Formal constraints for validation of defined components.

An LFB class can be seen in Figure 1. An LFB class has configurable components, read-only capabilities and subscribable events. Also an LFB class has one or more input ports where packets and/or metadata enter the LFB, and one or more output ports where packets and/or metadata exit the LFB. LFBs can be instantiated in the datapath.

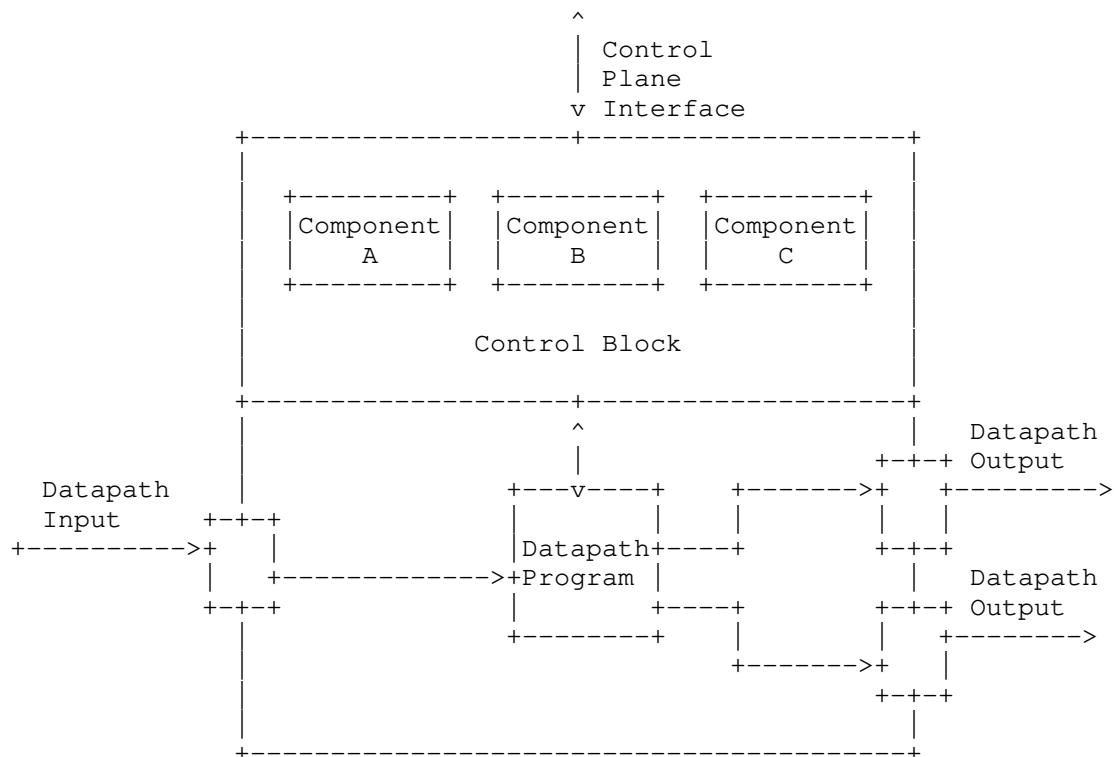


Figure 1: LFB Model

Most LFBs are related to packet processing. However there are cases such as discussed in [RFC5810] and [RFC5812] where two LFBs are defined, the FE Protocol LFB and the FE Object LFB respectively. The FE Protocol LFBs allows the control plane to configure, among others, the ForCES protocol mechanics, such as High availability and heartbeat mechanism. The FE Object LFB allows the control plane to configure, among others, the FE as whole, instantiate LFBs and manipulate the LFB service graph respectively.

### 3.3. ForCES & BNG

The ForCES architecture accrues several benefits. ForCES has the ability to add new packet services, described as LFB graphs, in an existing infrastructure. Secondly it can natively support any type of access, fixed, mobile, simply by modeling the necessary LFBs. One of the major advantages is that these abilities come with no change to the protocol; so long as the proper models (LFBs) are introduced, the ForCES protocol inherently supports them. To illustrate these claims, this document will start first by describing a basic

connectivity service and then augment that with a bandwidth management service.

For more details on how ForCES can support the separation of the forwarding from the control plane in regards to the BNG, the reader is encouraged to read the Control-Plane and User-Plane (CUSP) ForCES gap analysis [I-D.haleplidis-bcause-forces-gap-analysis] which elaborates on how ForCES meets the CUSP requirements as well as provide a ForCES XML model that detailed a CUSP information model. We hope to convince the reader that there already exists a robust IETF architecture which has a large deployment experience that meets all the CUSP requirements, while being more extensible and flexible with new requirements without any changes to the protocol.

#### 4. Basic BNG ForCES model

The BNG is the network edge aggregation point used by subscribers as the access point through which they connect to the broadband network. A subscriber could get access to the network using one or more different access types through the BNG. In this draft we focus on IPv4 PPPoE access.

Supporting different access types is easily done by adding appropriate LFBs that handle encapsulating and decapsulating the relevant protocol headers. As discussed in the previous section, adding new LFB models has no impact on the protocol itself. Future versions of this document will include additional access types.

A BNG is comprised of many different components. Figure 2 depicts components, modeled through ForCES, required to provide subscriber access using PPPoE. It is important to note that Figure 2 does not provide implementation details, but rather is a modelling of the underlying resources.

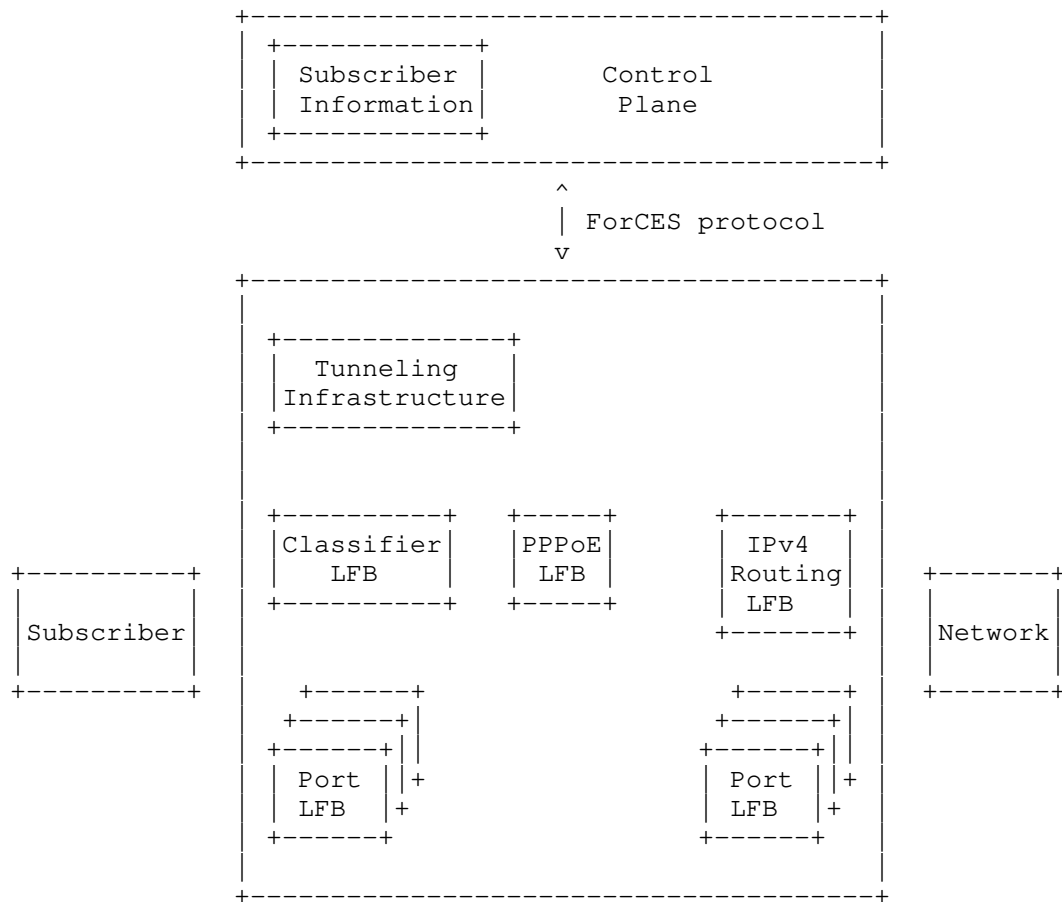


Figure 2: BNG LFB Components

There are a number of steps required for a subscriber to get access to the network. These include:

1. First the subscriber has to be authenticated. This will require a number of control packets to be exchanged between the subscriber and the control plane, redirected from the forwarding plane. For PPPoE the credentials are handed off to a RADIUS server; The communication between the control plane and the RADIUS server is out of scope of this document.
2. Once the subscriber is authenticated, the control plane has to configure the forwarding plane with the subscriber information, including resource allocation, authorizing the subscriber by providing access control and subscribed services.

3. Finally, as subscriber traffic is allowed through the BNG, the appropriate LFB instances are monitored by the control plane for accounting purposes either by polling or subscription to events. This would require either the control plane to poll the BNG or subscribe to accounting events from appropriate LFB instances.

In the first version of the draft, we focus on PPPoE as the access control mechanism. This first stage is the discovery and authentication phase via PPPoE and PPP.

#### 4.1. Authentication

There are two stages in PPPoE [RFC2516]. The Discovery stage and the PPP Session stage that will perform the discovery and authentication respectively.



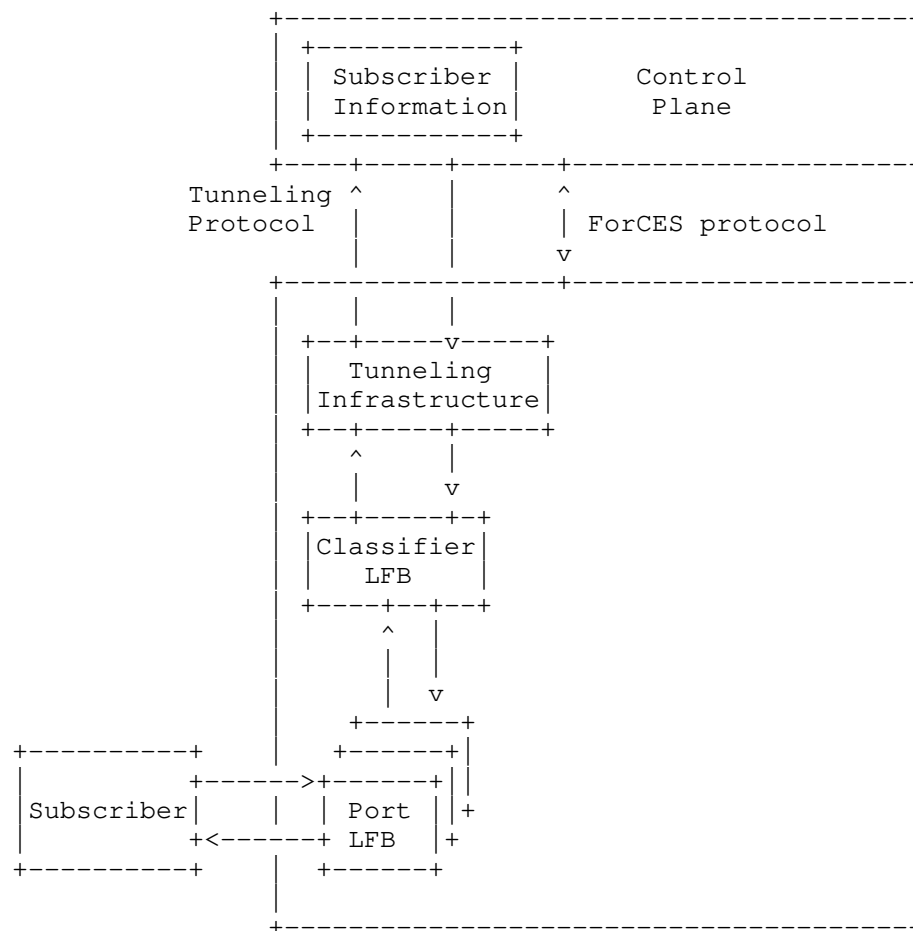


Figure 3: BNG Control Traffic handling

## 4.1.1.1. PPPoE Discovery stage

During the Discovery stage, frames arrive at an Ethernet Port (Port LFB). The frames are sent to the Classifier LFB as seen in Figure 3. The Classifier LFB determines whether these frames are control packets, distinguished by ethertype 0x8863. The control frames will be redirected via a tunneling infrastructure towards the control plane.

The tunneling infrastructure could be implemented by VxLAN, GRE, etc. ForCES could also be used in the case to redirect packets from the forwarding plane to the control plane. This is currently out of scope of this document.

The control plane handles the control messages and sends back the responses via the tunneling infrastructure to be redirected in the data path to be sent back to the subscriber.

Several control messages are exchanged in this stage between the subscriber and the controller - all recognized by the Classifier LFB, based on ethertype value 0x8863, and redirected to the control plane.

At the end of the PPPoE discovery stage both peers know the Session ID and the peer's MAC address which both identify uniquely the session.

At this stage the control plane knows the subscriber's Session ID and MAC address and programs the classifier.

#### 4.1.2. PPP Session stage

After the PPPoE Discovery stage has ended PPP traffic starts to flow. Frames arrive at the ethernet Port (LFB). The frames are sent to the Classifier LFB. PPPoE data packets are now distinguished by ethertype 0x8864 but there are still control packets passing prior to the subscriber being authenticated.

The Classifier LFB will distinguish control vs data packets, based on the protocol field of the encapsulated PPP header. Control packets such as LCP (0xC021), PAP (0xC023), CHAP (0xC223) and IPCP (0x8021), will be sent to the Control Plane to authenticate and authorize the Subscriber.

After the subscriber has been authenticated the authentication stage is finished, an IP address will be issued and the Subscriber will be considered authenticated and authorized.

#### 4.2. Subscriber Information Configuration

At this stage the controller is ready to configure the forwarding plane with the authorized subscriber information.

The control plane will create the basic connectivity service by configuring the PPP and the IPv4 Routing LFB to operate in forwarding mode in both the subscriber side direction as well as the internet side directions. The control plane will send a number of ForCES messages via the ForCES protocol to these LFBs with the parameters specific for the subscriber.

The Subscriber provisioned information, such as assigned IP, Session ID, service, MAC address, can reside either in the CE or in the FE as an individual LFB. Both options are valid in ForCES case. The

subscriber information, once set, will trigger a set of control commands to a number of LFBs; having it in the Control Plane will require more commands on the wire (ForCES messages) to be sent to the different LFBs. On the other hand, if the Subscriber Information rests in the data plane then less messaging is needed between control and data plane.

Once the subscriber information has been configured correctly, the subscriber's traffic starts passing through as illustrated in Figure 4. The Classifier LFB will distinguish PPPoE data packets by ethertype 0x8864, session ID, and IPv4 traffic by PPP protocol header 0x0021 and send them to the PPPoE LFB (with session ID as metadata) on the DecapIn input port of the PPPoE LFB.

The PPPoE LFB, once it validates the packet has arrived from an authorized subscriber based on the session ID and the MAC address, will decapsulate the IPv4 packet and send it to the IPv4 routing LFB, via the DecapOut output port.

The IPv4 routing LFB will route the packets and send it out the correct port LFB on the other side to be sent into the network.

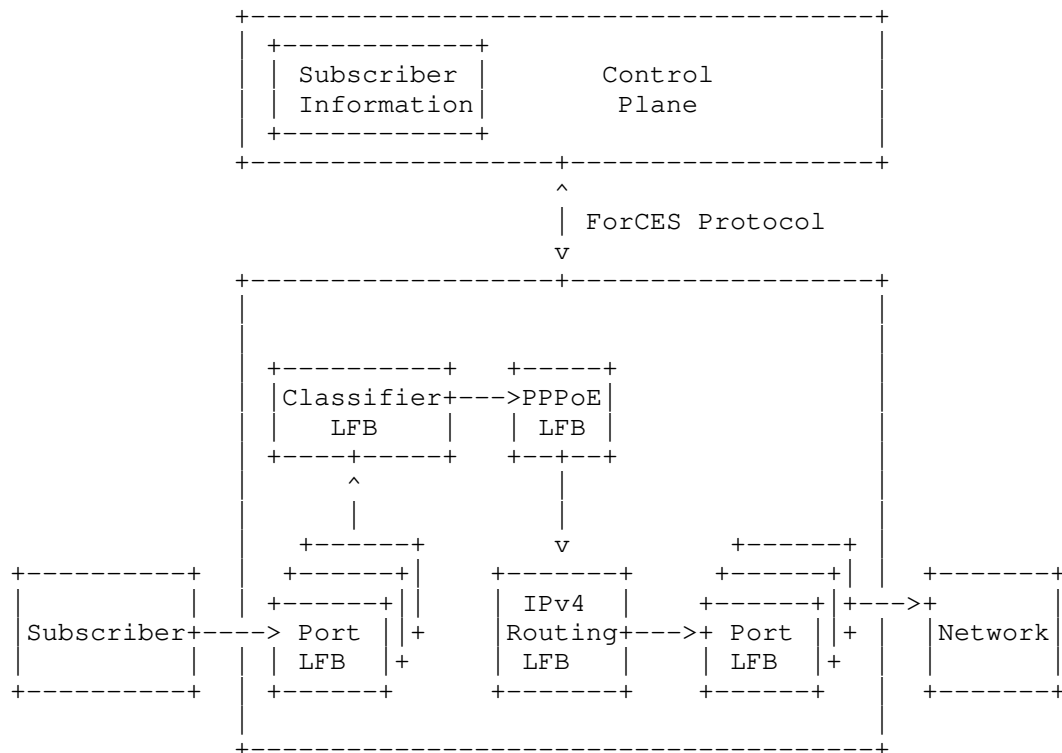


Figure 4: BNG Basic Connectivity Service (Upstream)

Figure 5 shows the downstream processing. When a packet is received from the network the port LFB will send the packet to the classifier LFB.

The classifier LFB will select the IPv4 traffic and generate a metadata (Subscriber ID) based on the destination IP address, to be propagated to downstream LFBs, and send the packets to the IPv4 Routing LFB. The IPv4 Routing LFB perform normal routing functions, such as selecting the correct output port and decreasing TTL. and pass on the packet to the PPPoE LFB through the EncapIn port.

The PPPoE LFB will encapsulate the packet with the correct PPPoE Session ID and destination MAC address and send the traffic back to the subscriber via the EncapOut port to the Port LFB with the interface index provided by the IPv4 routing LFB.

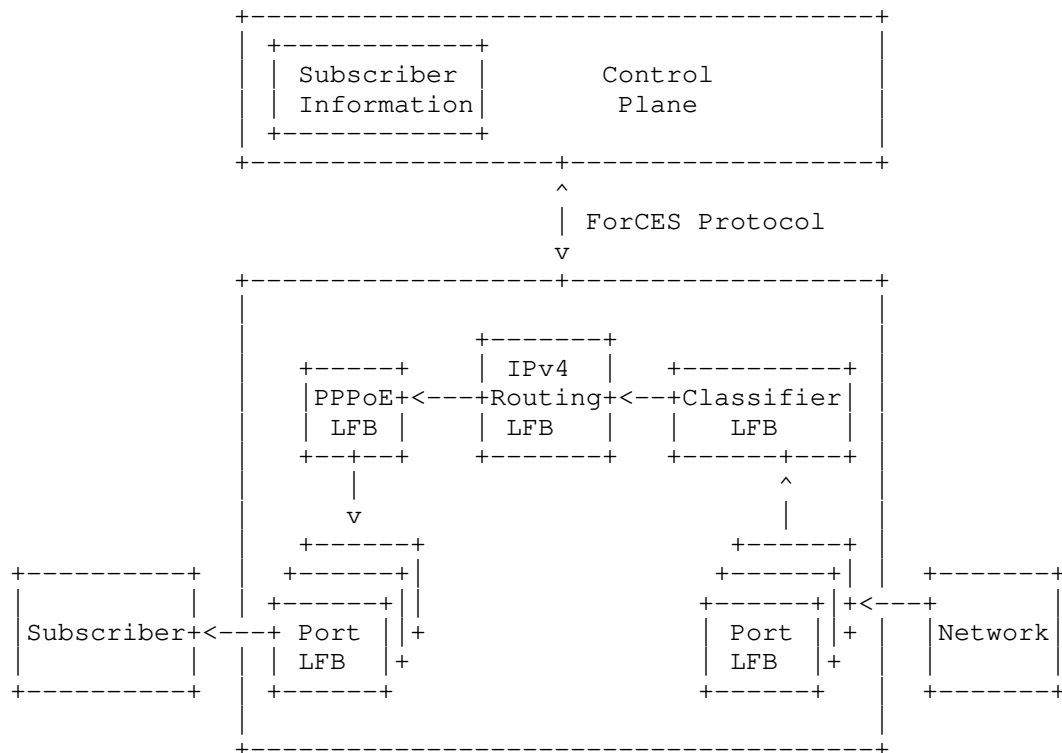


Figure 5: BNG Basic Connectivity Service (Downstream)

#### 4.2.1. Supporting multiple access types

As has been discussed earlier in the document, any kind of access mechanism can be supported by the ForCES model, whether that is fixed access, or mobile or wireless, or optical.

A relevant port LFB instantiation will be needed to handle traffic entering the BNG

The classifier can be augmented to distinguish the packets based on their access mechanism, e.g. ethertypes for ethernet based access types and will send the packets to the relevant LFBs to be handled.

This has no effect on the ForCES protocol. So long as everything has been modeled with the ForCES model, the protocol will be able to control and manage the LFBs with no change.

#### 4.3. Traffic monitoring

As soon as Subscriber traffic starts flowing through the BNG, it is imperative to monitor for accounting purposes.

Subscriber usage monitoring can be achieved in a number of ways, with the easiest being monitoring incoming and outgoing statistics at the PPPoE LFB. The controller can either poll the LFB for said statistics, or it can subscribe to events to the PPPoE LFB and receive notifications for statistics.

#### 5. Advanced BNG Services

The BNG is composed of many more functions and components. ForCES provides the ability to provision new services.

A service, in ForCES terms, is a graph of LFBs. Figure 4 and Figure 5 showcase a simple connectivity service. The operator via the control plane can provision new services by creating new graphs with existing LFBs or introducing new LFBs and composing new LFB graphs. One important detail in ForCES, as has been discussed before, is that defining or adding new LFBs has absolutely no impact on the protocol itself.

##### 5.1. Bandwidth Management Service

A service provider can introduce a bandwidth management service with no extension to the protocol. Adding rate limiting both in the upstream to the internet and downstream from the internet is simply done by adding a new LFB called policer as can be seen in Figure 6 and Figure 7. A service provider could add an instance for each subscriber to give them different bandwidth rates based on SLAs. Example 1 mbps up/3 mbps down vs 3 mbps up and 8mbps down.

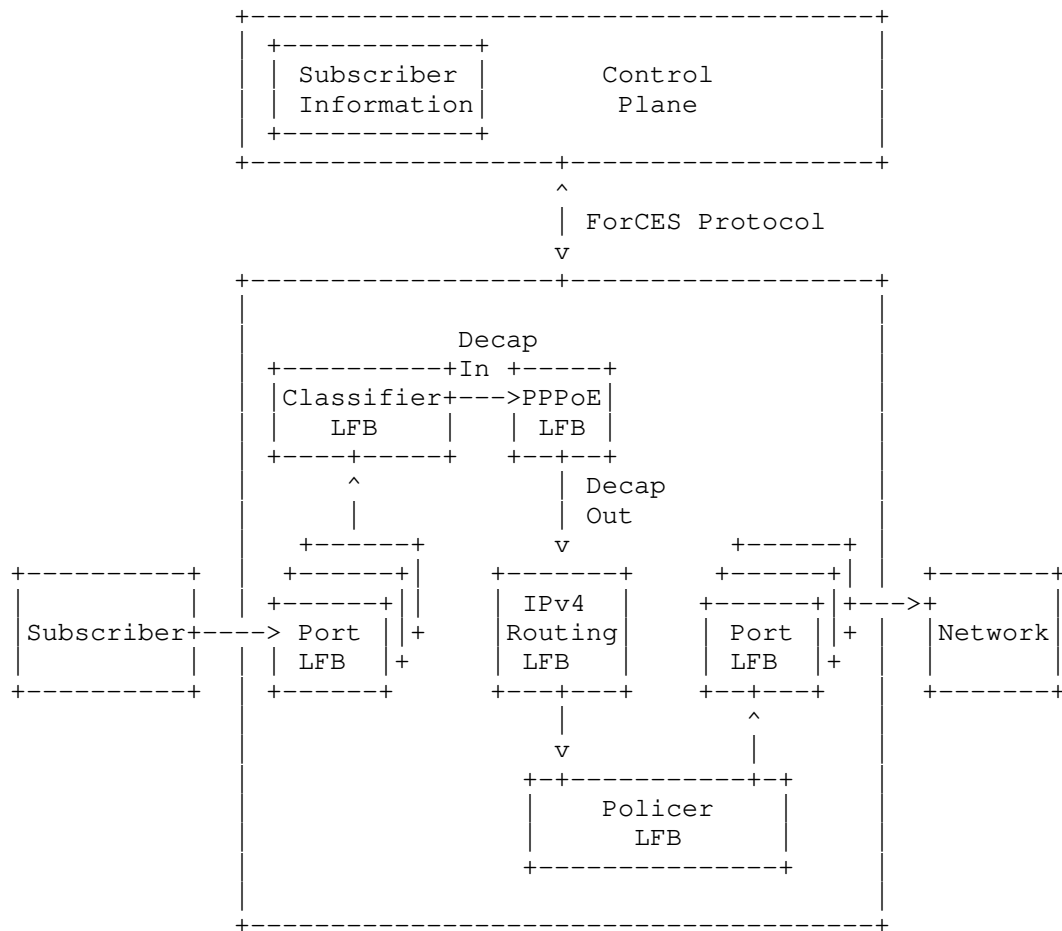


Figure 6: BNG Bandwidth management service (Upstream)

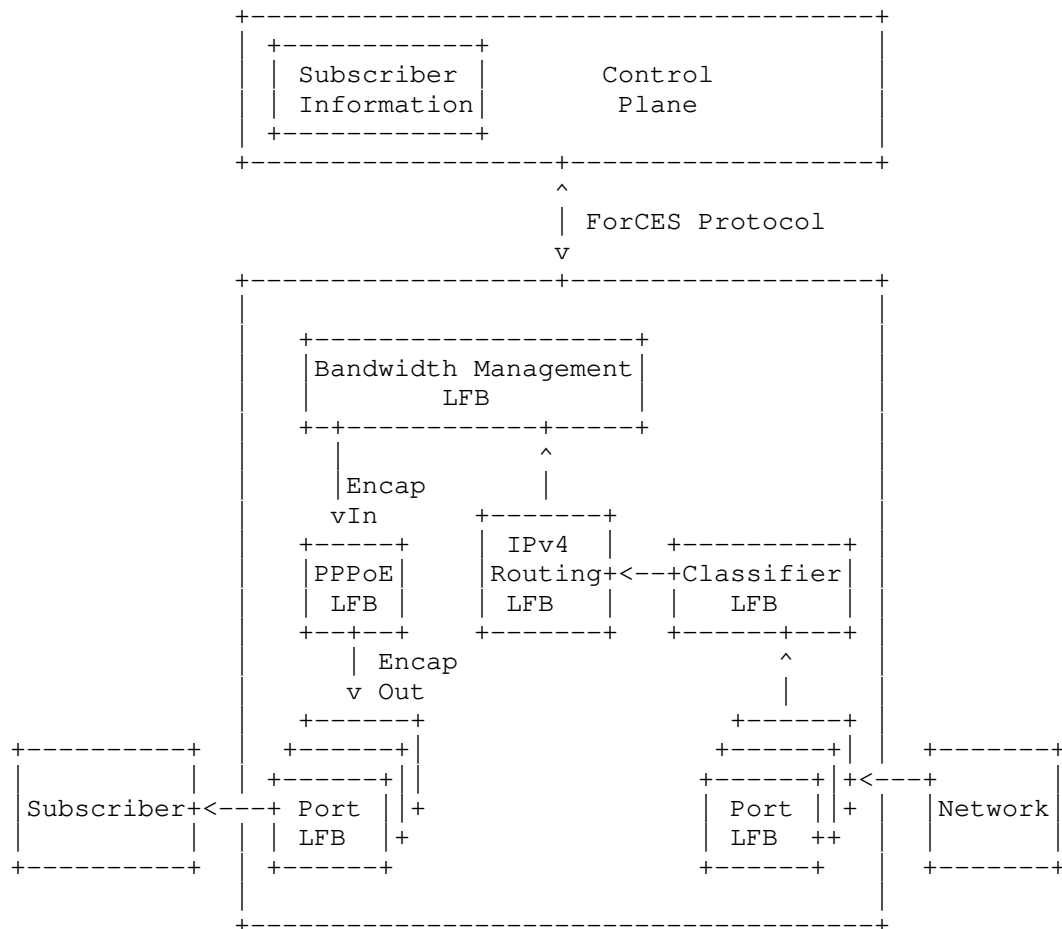


Figure 7: BNG Bandwidth management service (Downstream)

## 5.2. Stateless access control service

In case an operator requires services like a firewall, an Access Control List (ACL) LFB can be instantiated and inserted into the graph to drop or allow packets.

## 5.3. Quota Enforcement service

In case an operator require to perform quota enforcement, a Quota Enforcement (QE) LFB can be instantiated and inserted into the graph to drop or allow packets depending on the Subscriber SLA.



#### 5.4. Lawful Intercept service

In the case of creating a Lawful Intercept, an operator simply by instantiating a mirror LFB into the datapath to create copies of the packets to be sent to a specific destination

### 6. LFB Class Descriptions

As has been discussed before, an LFB is a well defined logical functional block residing in the FE and is an accurate abstraction of the FE's processing capabilities.

This document provides a sample LFB class library pertaining to the separation of the forwarding and the control plane for the BNG. [RFC6956] provides an additional LFB class library for a typical router.

#### 6.1. Port LFB

The Port LFB abstracts all the interfaces, physical and virtual, in the device. The LFB handles frames coming in from or out of the FE. The current port class LFB is defined to receive ethernet frames.

##### 6.1.1. Data Handling

When frames arrive from the Subscriber or the Network side, these frames are received by the Port LFB via IngressInPort, the physical port of the BNG, to be passed downstream to the Classifier LFB via the IngressOutPort.

When frames are meant to be sent to the Subscriber or the Network side, these frames arrive to the Port LFB via the EgressInPort alongside an ifIndex metadata to notify which interface should be used, and sent out via the EgressOutPort.

Figure 8 illustrates the Port LFB

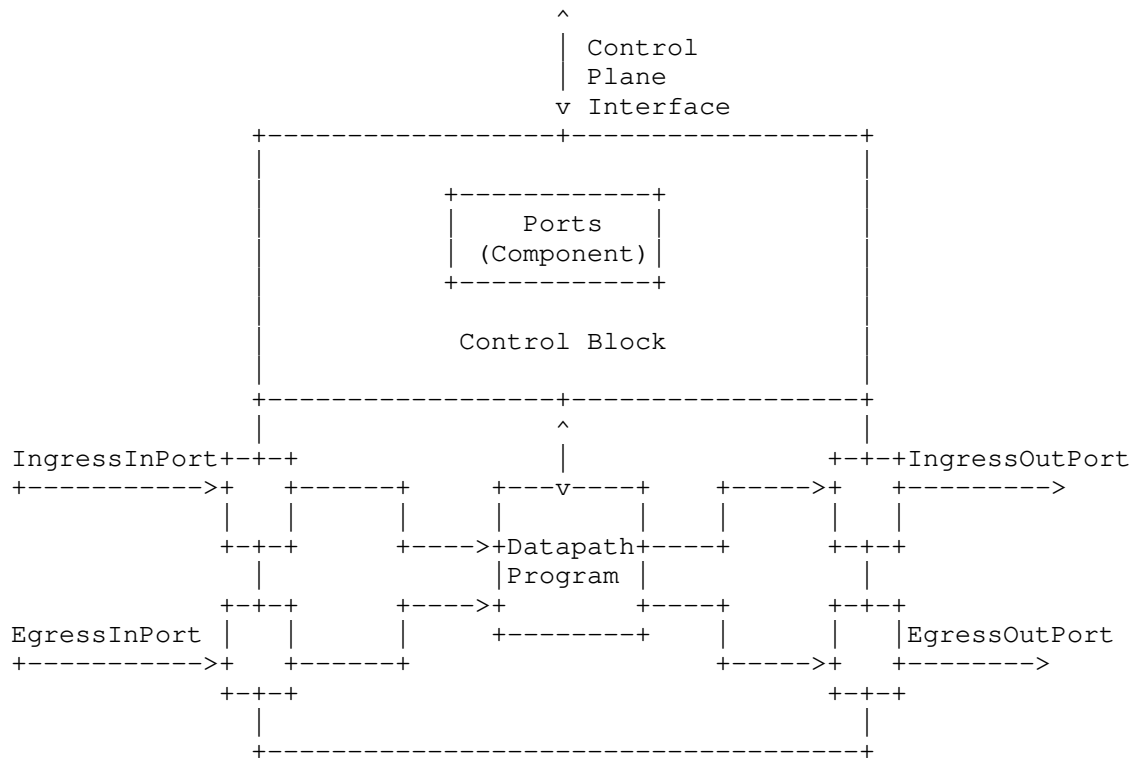


Figure 8: Port LFB

#### 6.1.2. Components

This LFB has defined only one component, named ports. The ports component is table. Each entry in the table contains the parameters for a port within the BNG. All the entries consist of all the ports of the BNG. Each table entry is of type PortInfo.

The PortInfo type contains the following configurable parameters:

Name - A string representation for the name of the port

ifindex - the interface index of the port

L2Address - The MAC address of the port

MTU - The maximum transmit unit for this port

Stats - 64-bit counters for statistics for the port. These include Tx,Rx bytes and packets, errors and dropped packets.

Operstate - The Operational state of the port, such as down, up

Adminstate - The Administrative state of the port, such as down, up

Promiscuity - The port promiscuity

#### 6.1.3. Capabilities

This LFB does not have a list of capabilities.

#### 6.1.4. Events

This LFB has four events specified:

Port changed - This event will trigger when an existing port has been updated.

Port deleted - This event will trigger when an existing port has been deleted such as deleting a virtual port or removing a physical interface.

Port created - This event will trigger when an new port has been created such as creating a virtual port or inserting a physical interface.

Port stats changed - This event will trigger periodically when stats change. It is important to properly configure the event parameters (hysteresis, interval) to avoid generation of multiple event notifications

#### 6.2. Classifier LFB

The classifier LFB abstracts the classification of packets into different categories. From the subscriber side to the network, it is required for the classifier LFB to detect Subscriber control traffic to be redirected to the control plane, as well as detect Subscriber data traffic to be sent downstream with the subscriber ID as metadata. From the network side to the subscriber, it is required for the classifier LFB to detect the Subscriber ID from the destination IP address and send the packet with the subscriber ID as metadata to the next LFB. In addition the classifier LFB has to discriminate control packets arriving from the control plane via the tunneling infrastructure which have to be sent back to the subscriber.

### 6.2.1. Data Handling

Packets enter the classifier LFB via the InPort. Ethernet packets arriving from the subscriber side with Ethertype 0x8863 are considered PPPoE control messages and are sent to the control plane via the RedirectOut output port. Similarly, ethernet frames with EtherType 0x8864 but the PPP protocol field belonging to a control protocol, will be sent to the control plane via the ControlOut output port

Packets returning via the tunneling infrastructure enter the classifier LFB via the InPort, are matched based on destination MAC address, Ethertype and optionally PPP protocol field and are sent to the Port LFB via the EthernetOut port to be sent to the subscriber.

Subscriber data traffic coming from the subscriber's side are matched based on EtherType 0x8864 and PPP protocol 0x0021 (IPv4) and also based on the PPPoE session ID and Subscriber MAC address. Once matched, the classifier will issue the Subscriber ID (configured by the control plane) and pass it along the packet as metadata to the next LFB (PPPoE LFB) via the EthernetOut port.

Subscriber data traffic coming from the network's side are matched based on IPv4 destination address. Once matched, they will be issued the Subscriber ID (configured by the control plane) and pass it along to the next LFB (IPv4 Routing) via the IPv4Out port.

### 6.2.2. Components

This LFB has one component, the filters which is the table that contains all the classification filters.

Each entry in the table contains three fields:

Filter Name - The name of the filter

Actions - A table of actions to be performed on the packet

Filter keys - A table of key values to be matched on the packets

In regards to the Actions, each table entry of the actions contains two fields:

Action Type - The name of the action

Actions - The action itself. Currently only three actions have been defined, drop forward and redirect to the controller

In regards to the Filter keys, each table entry of the filter keys contains five optional fields:

IP protocol - Detects a specific transport protocol

Ethernet - Matches on specific ethernet fields

IP - Matches on specific IPv4 header fields

PPP Control - To detect and match specific PPP control messages

PPP Subscriber traffic - To detect PPP subscriber data traffic

#### 6.2.3. Capabilities

This LFB does not have a list of capabilities.

#### 6.2.4. Events

This LFB has three events specified:

Filter Changed - An Existing filter has been changed.

Filter deleted - This event will trigger when an existing filter has been deleted.

Filter created - This event will trigger when a new filter has been created.

#### 6.3. PPPoE LFB

The PPPoE LFB is responsible for Encapsulating IPv4 Packets into PPPoE Ethernet frames coming from the network side towards the subscriber, and decapsulating IPv4 Packets from PPPoE frames arriving from the subscriber to be sent to the network.

##### 6.3.1. Data Handling

This LFB handles traffic differently depending on whether it needs to be encapsulated into an PPPoE frame or decapsulate a PPPoE frame and the IPv4 packet from within.

Regarding encapsulation, IPv4 packets arrive from the network side via the EncapIn port alongside the subscriber ID generated by the classifier LFB and the OutIfIndex metadata generated by the IPv4 routing LFB. The LFB looks up the subscriber information and creates the PPPoE frame based on the information located in the PPPoE Subscriber Information table entry. The generated frame is sent to

the Port LFB via the EncapOut port to the port specified by the OutIfIndex.

Regarding decapsulation, PPPoE frame arrive from the subscriber side via the DecapIn port alongside the subscriber ID generated by the classifier LFB. The LFB looks up the subscriber information and decapsulates the IPv4 packet within. The resulting IPv4 packet is sent to the IPv4 Routing LFB via the DecapOut port.

#### 6.3.2. Components

This LFB has two components, the first is the PPPoE Subscriber Information and the second are the Encap and Decap Statistics for encapsulating and decapsulating packets. Both components are tables.

Each entry in the PPPoE Subscriber Information table contains the following:

Subscriber ID - The subscriber identifier. This value is also used to look up table entries.

PPPoE Session ID - The Session ID of the Subscriber

Subscriber MAC Address - The Subscriber's MAC Address

Local MAC Address - The MAC Address of the local interface connected to the Subscriber.

MSS - The Maximum Segment Size

MRU - The Maximum Receive Unit

Magic Number - The magic number

Peer Magic Number - The subscriber's magic number

Each entry in the Statistics table contains the following:

Subscriber ID - The subscriber identifier. This value is also used to look up table entries.

Rx Encap Packets - The number of packets received at the encap ingress side

Tx Encap Packets - The number of packets transmitted at the encap egress side

Rx Encap Bytes - The number of bytes received at the encap ingress side

Tx Encap Bytes - The number of bytes transmitted at the encap egress side

Rx Decap Packets - The number of packets received at the decap ingress side

Tx Decap Packets - The number of packets transmitted at the decap egress side

Rx Decap Bytes - The number of bytes received at the decap ingress side

Tx Decap Bytes - The number of bytes transmitted at the decap egress side

#### 6.3.3. Capabilities

This LFB does not have a list of capabilities.

#### 6.3.4. Events

This LFB has four events specified:

Subscriber Changed - Existing PPPoE subscriber information has been changed.

Subscriber deleted - This event will trigger when an existing PPPoE subscriber information has been deleted.

Subscriber created - This event will trigger when a new PPPoE subscriber information has been created.

Stats changed - This event will trigger periodically when stats change. It is important to properly configure the event parameters (hysteresis, interval) to avoid generation of multiple event notifications

#### 6.4. IPv4 Routing LFB

The model of IPv4 Routing LFB is responsible for selecting the next hop for upstream and downstream traffic and then to generate the correct MAC address to be inserted into the destination MAC address and the ifindex of the output port so that the packet can be send out of the BNG.

#### 6.4.1. Data Handling

This LFB is used for packets sent from the Subscriber to the Network and back to locate the next hop and select the output port. IPv4 packets enter this LFB via the InPort, alongside the Subscriber Index as metadata. The IPv4 Routing LFB performs a lookup using the destination IP address on the IPv4Routing table. Once an entry has been found, the packet is checked for MTU and the next hop index is found. The next hop index is used to locate the correct entry in the IPv4NextHopTable. Once the entry has been found, the IPv4 Routing LFB decrements the TTL and populates the destination MAC address and sends the packet to the Port LFB with the ifindex metadata specifying which output port is going to be used via the NormalOut port. If an error occurs in the checksum, or TTL, packets are dropped and exit the LFB via the ExceptionOut port.

#### 6.4.2. Components

This LFB has two components, the first is the IPv4Routing Table which is the table for the IPv4 Longest Prefix Match and the second is the IPv4NextHopTable which contains information required for the next hop. Both components are tables.

Each entry in the IPv4Routing table contains the following:

Subscriber ID - The subscriber identifier.

IPv4 Address - The destination IPv4 address which is used to look up table entries.

Prefix Length - The prefix length

Hop Selector - Index for the Next Hop Table to lookup the Next hop information.

Each entry in the IPv4NextHopTable table contains the following:

OutIfIndex - This is the interface index out of which this packet should be sent.

MTU - The MTU of the outgoing port.

Next Hop IP Address - The next hop IPv4 Address

Next Hop MAC Address - The next hop MAC Address.



#### 6.4.3. Capabilities

This LFB does not have a list of capabilities.

#### 6.4.4. Events

This LFB does not have a list of events.

### 6.5. Policer LFB

The Policer LFB is responsible for maintaining the QoS requirements for a specific user based on his SLA either for upstream or for downstream traffic. Currently only four characteristics have been defined. The four characteristics are the Committed Information Rate, the Peak Information Rate, the Committed Burst Size and the Peak Burst Size. This LFB can be further augmented with more QoS parameters.

#### 6.5.1. Data Handling

For upstream this LFB receives IPv4 packets and Subscriber ID as metadata at the InUpstreamPort and based on the QoS parameters for that specific Subscriber, manipulates the packets to maintain the agreed upon SLA. Output packets are sent out via the OutUpstreamPort alongside the metadata, Subscriber ID.

Similarly for downstream traffic, this LFB receives IPv4 packets and Subscriber ID as metadata at the InDownstreamPort and based on the QoS parameters for that specific Subscriber, manipulates the packets to maintain the agreed upon SLA. Output packets are sent out via the OutDownstreamPort alongside the metadata, Subscriber ID.

#### 6.5.2. Components

This LFB has been defined with two components, one for upstream and one for downstream traffic named PoliceUpstream and PoliceDownstream respectively. Both components are tables. Each entry in each table contains the four QoS parameters and the Subscriber ID. Each table entry is of type PolicerTableEntry.

The PolicerTableEntry type contains the following configurable parameters:

SubscriberID - A uint32 value to identify the Subscriber

CIR - a uint32 value to represent the Committed Information Rate in kilobits per second.

PIR - a uint32 value to represent the Peak Information Rate in kilobits per second.

CBS - a uint32 value to represent the Committed Burst Size in bytes.

PBS - a uint32 value to represent the Peak Burst Size in bytes.

### 6.5.3. Capabilities

This LFB does not have a list of capabilities.

### 6.5.4. Events

This LFB does not have a list of events.

## 7. BNG ForCES XML model

In this section we provide ForCES based XML models for the LFBs in the BNG ForCES model

```
<?xml version="1.0" encoding="UTF-8"?>
<LFBLibrary xmlns="urn:ietf:params:xml:ns:forces:lfbmodel:1.1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
provides="vbng">
  <frameDefs>
    <frameDef>
      <name>Arbitrary</name>
      <synopsis>Any kind of packet</synopsis>
    </frameDef>
    <frameDef>
      <name>EthernetFrame</name>
      <synopsis>An ethernet frame</synopsis>
    </frameDef>
    <frameDef>
      <name>IPv4Packet</name>
      <synopsis>An IPv4 packet</synopsis>
    </frameDef>
  </frameDefs>
  <dataTypeDefs>
    <!-- For Port LFB -->
    <dataTypeDef>
      <name>operstates</name>
      <synopsis>
        The possible operational states of a port link (RFC 2863)
      </synopsis>
      <atomic>
        <baseType>uchar</baseType>
      </atomic>
    </dataTypeDef>
  </dataTypeDefs>
</LFBLibrary>
```

```

<specialValues>
  <specialValue value="0">
    <name>OS_UNKNOWN</name>
    <synopsis>Unknown value</synopsis>
  </specialValue>
  <specialValue value="1">
    <name>OS_NOTPRESENT</name>
    <synopsis>The link is not present</synopsis>
  </specialValue>
  <specialValue value="2">
    <name>OS_DOWN</name>
    <synopsis>The link is operationally down</synopsis>
  </specialValue>
  <specialValue value="3">
    <name>OS_LOWERLAYERDOWN</name>
    <synopsis>The link of the lower port is down</synopsis>
  </specialValue>
  <specialValue value="4">
    <name>OS_TESTING</name>
    <synopsis>The Link is undergoing some testing</synopsis>
  </specialValue>
  <specialValue value="5">
    <name>OS_DORMANT</name>
    <synopsis>Link is in the dormant state</synopsis>
  </specialValue>
  <specialValue value="6">
    <name>OS_UP</name>
    <synopsis>The Link is operationally up</synopsis>
  </specialValue>
</specialValues>
</atomic>
</dataTypeDef>
<dataTypeDef>
  <name>adminstates</name>
  <synopsis>
    The possible administrative states of a port link (RFC 2863)
  </synopsis>
<atomic>
  <baseType>uchar</baseType>
  <specialValues>
    <specialValue value="1">
      <name>AS_DOWN</name>
      <synopsis>The link is operationally down</synopsis>
    </specialValue>
    <specialValue value="2">
      <name>AS_LOWERLAYERDOWN</name>
      <synopsis>The link of the lower port is down</synopsis>
    </specialValue>
  </specialValues>

```

```
<specialValue value="3">
  <name>AS_DORMANT</name>
  <synopsis>Link is in the dormant state</synopsis>
</specialValue>
<specialValue value="4">
  <name>AS_UP</name>
  <synopsis>The Link is operationally up</synopsis>
</specialValue>
</specialValues>
</atomic>
</dataTypeDef>
<dataTypeDef>
  <name>devstats</name>
  <synopsis>Port stats</synopsis>
  <struct>
    <component componentID="1">
      <name>rx_packets</name>
      <synopsis>Total packets received</synopsis>
      <typeRef>uint64</typeRef>
    </component>
    <component componentID="2">
      <name>tx_packets</name>
      <synopsis>Total packets transmitted</synopsis>
      <typeRef>uint64</typeRef>
    </component>
    <component componentID="3">
      <name>rx_bytes</name>
      <synopsis>Total bytes received</synopsis>
      <typeRef>uint64</typeRef>
    </component>
    <component componentID="4">
      <name>tx_bytes</name>
      <synopsis>Total bytes transmitted</synopsis>
      <typeRef>uint64</typeRef>
    </component>
    <component componentID="5">
      <name>rx_errors</name>
      <synopsis>Total packet receive errors</synopsis>
      <typeRef>uint64</typeRef>
    </component>
    <component componentID="6">
      <name>tx_errors</name>
      <synopsis>Total packet transmit errors</synopsis>
      <typeRef>uint64</typeRef>
    </component>
    <component componentID="7">
      <name>rx_dropped</name>
      <synopsis>
```

```

    Total packet received and dropped. Typically because no
    space
  </synopsis>
    <typeRef>uint64</typeRef>
  </component>
  <component componentID="8">
    <name>tx_dropped</name>
    <synopsis>
      Total packet transmit dropped Typically because no space
    </synopsis>
    <typeRef>uint64</typeRef>
  </component>
  <component componentID="9">
    <name>multicast</name>
    <synopsis>Total multicast packets received</synopsis>
    <typeRef>uint64</typeRef>
  </component>
  <component componentID="10">
    <name>tx_collisions</name>
    <synopsis>Total transmit packet collisions on the
      link</synopsis>
    <optional/>
    <typeRef>uint64</typeRef>
  </component>
  <component componentID="11">
    <name>rx_length_errors</name>
    <synopsis>rx errors because of length mismatch</synopsis>
    <optional/>
    <typeRef>uint64</typeRef>
  </component>
  <component componentID="12">
    <name>rx_over_errors</name>
    <synopsis>rx errors because of buffer overflows</synopsis>
    <optional/>
    <typeRef>uint64</typeRef>
  </component>
  <component componentID="13">
    <name>rx_crc_errors</name>
    <synopsis>rx errors because of crc errors</synopsis>
    <optional/>
    <typeRef>uint64</typeRef>
  </component>
  <component componentID="14">
    <name>rx_frame_errors</name>
    <synopsis>rx errors because of frame alignment
      error</synopsis>
    <optional/>
    <typeRef>uint64</typeRef>
  </component>

```

```
</component>
<component componentID="15">
  <name>rx_fifo_errors</name>
  <synopsis>rx errors because of fifo overruns</synopsis>
  <optional/>
  <typeRef>uint64</typeRef>
</component>
<component componentID="16">
  <name>rx_missed_errors</name>
  <synopsis>rx errors because of missed packets</synopsis>
  <optional/>
  <typeRef>uint64</typeRef>
</component>
<component componentID="17">
  <name>tx_aborted_errors</name>
  <synopsis>tx errors because of tx abort</synopsis>
  <optional/>
  <typeRef>uint64</typeRef>
</component>
<component componentID="18">
  <name>tx_carrier_errors</name>
  <synopsis>tx errors because of carrier problems</synopsis>
  <optional/>
  <typeRef>uint64</typeRef>
</component>
<component componentID="19">
  <name>tx_fifo_errors</name>
  <synopsis>tx errors because of fifo problems</synopsis>
  <optional/>
  <typeRef>uint64</typeRef>
</component>
<component componentID="20">
  <name>tx_heartbeat_errors</name>
  <synopsis>tx errors because of heartbeat problems</synopsis>
  <optional/>
  <typeRef>uint64</typeRef>
</component>
<component componentID="21">
  <name>tx_window_errors</name>
  <synopsis>tx errors because of windowing problems</synopsis>
  <optional/>
  <typeRef>uint64</typeRef>
</component>
<component componentID="22">
  <name>rx_compressed</name>
  <synopsis>Total rx compressed packets</synopsis>
  <optional/>
  <typeRef>uint64</typeRef>
```

```

    </component>
    <component componentID="23">
      <name>tx_compressed</name>
      <synopsis>Total tx compressed packets</synopsis>
      <optional/>
      <typeRef>uint64</typeRef>
    </component>
  </struct>
</dataTypeDef>
<dataTypeDef>
  <name>PortInfo</name>
  <synopsis>Describing the Port Details</synopsis>
  <struct>
    <component componentID="1">
      <name>name</name>
      <synopsis>The name of the port</synopsis>
      <optional/>
      <typeRef>string[16]</typeRef>
    </component>
    <component componentID="2">
      <name>ifindex</name>
      <synopsis>The ifindex of the port</synopsis>
      <typeRef>uint32</typeRef>
    </component>
    <component componentID="3">
      <name>L2Address</name>
      <synopsis>The MAC address</synopsis>
      <optional/>
      <typeRef>byte[6]</typeRef>
    </component>
    <component componentID="4">
      <name>mtu</name>
      <synopsis>The Maximum transmit unit for this port</synopsis>
      <optional/>
      <typeRef>uint32</typeRef>
    </component>
    <component componentID="5">
      <name>flags</name>
      <synopsis>
flags for config and operational state. On the FE CE
direction, these flags depend on flags mask to point to
which flags to change
      </synopsis>
      <optional/>
      <typeRef>uint32</typeRef>
    </component>
    <component componentID="6">
      <name>flagsmask</name>

```

```

    <synopsis>
Mask for flags for config and operational state In config
direction, a bit turned on indicates that the FE is to set
the corresponding flags to value specified
</synopsis>
    <optional/>
    <typeRef>uint32</typeRef>
</component>
<component componentID="7">
    <name>stats</name>
    <synopsis>The 64-bit port stats</synopsis>
    <optional/>
    <typeRef>devstats</typeRef>
</component>
<component componentID="8">
    <name>operstate</name>
    <synopsis>The Link operational state of the port</synopsis>
    <optional/>
    <typeRef>operstates</typeRef>
</component>
<component componentID="9">
    <name>operstate</name>
    <synopsis>The Link operational state of the port</synopsis>
    <optional/>
    <typeRef>adminstates</typeRef>
</component>
<component componentID="10">
    <name>promiscuity</name>
    <synopsis>The port promiscuity</synopsis>
    <optional/>
    <typeRef>uint32</typeRef>
</component>
</struct>
</dataTypeDef>
<!-- For Classifier -->
<dataTypeDef>
    <name>ethaddrs_key</name>
    <synopsis>Ethernet address key structure</synopsis>
    <struct>
        <component componentID="1">
            <name>eth_dst</name>
            <synopsis>Destination Ethernet address</synopsis>
            <optional/>
            <typeRef>byte[6]</typeRef>
        </component>
        <component componentID="2">
            <name>eth_dst_mask</name>
            <synopsis>Destination Ethernet address mask</synopsis>

```



```
        <optional/>
        <typeRef>byte[6]</typeRef>
    </component>
    <component componentID="3">
        <name>eth_src</name>
        <synopsis>Source Ethernet address</synopsis>
        <optional/>
        <typeRef>byte[6]</typeRef>
    </component>
    <component componentID="4">
        <name>eth_src_mask</name>
        <synopsis>Source Ethernet address mask</synopsis>
        <optional/>
        <typeRef>byte[6]</typeRef>
    </component>
</struct>
</dataTypeDef>
<dataTypeDef>
    <name>inaddr_key</name>
    <synopsis>IPv4 Address key structure</synopsis>
    <struct>
        <component componentID="1">
            <name>src</name>
            <synopsis>IP Source address (BE)</synopsis>
            <optional/>
            <typeRef>octetstring[4]</typeRef>
        </component>
        <component componentID="2">
            <name>src_mask</name>
            <synopsis>IP Source address mask (BE)</synopsis>
            <optional/>
            <typeRef>octetstring[4]</typeRef>
        </component>
        <component componentID="3">
            <name>dst</name>
            <synopsis>IP Destination address (BE)</synopsis>
            <optional/>
            <typeRef>octetstring[4]</typeRef>
        </component>
        <component componentID="4">
            <name>dst_mask</name>
            <synopsis>IP Destination address mask (BE)</synopsis>
            <optional/>
            <typeRef>octetstring[4]</typeRef>
        </component>
    </struct>
</dataTypeDef>
<dataTypeDef>
```

```
<name>PPPoE_control_key</name>
<synopsis>PPPoE control key structure</synopsis>
<struct>
  <component componentID="1">
    <name>PPPoEControl</name>
    <synopsis>PPPoE Control Traffic. Default 0x8863</synopsis>
    <optional/>
    <typeRef>uint16</typeRef>
  </component>
  <component componentID="2">
    <name>PPPControl</name>
    <synopsis>PPP Control Protocol Traffic</synopsis>
    <optional/>
    <typeRef>uint16</typeRef>
  </component>
</struct>
</dataTypeDef>
<dataTypeDef>
  <name>PPPoE_subscriber_key</name>
  <synopsis>PPPoE control key structure</synopsis>
  <struct>
    <component componentID="1">
      <name>PPPSessionID</name>
      <synopsis>Session ID</synopsis>
      <optional/>
      <typeRef>uint16</typeRef>
    </component>
    <component componentID="2">
      <name>SubMACAddress</name>
      <synopsis>Session ID</synopsis>
      <optional/>
      <typeRef>octetstring[6]</typeRef>
    </component>
  </struct>
</dataTypeDef>
<dataTypeDef>
  <name>keyinfo</name>
  <synopsis>Describes the BNG classifier key</synopsis>
  <struct>
    <component componentID="1">
      <name>ip_proto</name>
      <synopsis>Transport protocols: TCP, UDP, SCTP, ICMP,
        ICMPV6</synopsis>
      <optional/>
      <atomic>
        <baseType>uchar</baseType>
        <specialValues>
          <specialValue value="1">
```

```
        <name>IPPROTO_ICMP</name>
        <synopsis/>
    </specialValue>
    <specialValue value="6">
        <name>IPPROTO_TCP</name>
        <synopsis/>
    </specialValue>
    <specialValue value="17">
        <name>IPPROTO_UDP</name>
        <synopsis/>
    </specialValue>
    <specialValue value="132">
        <name>IPPROTO_SCTP</name>
        <synopsis/>
    </specialValue>
    <specialValue value="58">
        <name>IPPROTO_ICMPV6</name>
        <synopsis/>
    </specialValue>
</specialValues>
</atomic>
</component>
<component componentID="2">
    <name>eth</name>
    <synopsis>Ethernet header key</synopsis>
    <optional/>
    <typeRef>ethhdrs_key</typeRef>
</component>
<component componentID="3">
    <name>ip</name>
    <synopsis>IP header key</synopsis>
    <optional/>
    <typeRef>inaddr_key</typeRef>
</component>
<component componentID="4">
    <name>PPPCControl</name>
    <synopsis>PPPoE control headers</synopsis>
    <optional/>
    <typeRef>PPPoE_control_key</typeRef>
</component>
<component componentID="5">
    <name>PPPSubscriberTraffic_key</name>
    <synopsis>PPPoE data headers</synopsis>
    <optional/>
    <typeRef>PPPoE_subscriber_key</typeRef>
</component>
</struct>
</dataTypeDef>
```

```
<dataTypeDef>
  <name>filteractinfo</name>
  <synopsis>Basic filter action row entry</synopsis>
  <struct>
    <component componentID="1">
      <name>factype</name>
      <synopsis>Action type name</synopsis>
      <typeRef>string[16]</typeRef>
    </component>
    <component componentID="2">
      <name>faction</name>
      <synopsis>Action</synopsis>
      <atomic>
        <baseType>uchar</baseType>
        <specialValues>
          <specialValue value="0">
            <name>Drop</name>
            <synopsis>Drop packet</synopsis>
          </specialValue>
          <specialValue value="1">
            <name>ForwardPacket</name>
            <synopsis>Forward packet</synopsis>
          </specialValue>
          <specialValue value="2">
            <name>Redirect</name>
            <synopsis>Redirect packet to controller</synopsis>
          </specialValue>
        </specialValues>
      </atomic>
    </component>
  </struct>
</dataTypeDef>
<dataTypeDef>
  <name>ClassifierFilterInfo</name>
  <synopsis>Basic filter row entry</synopsis>
  <struct>
    <component componentID="1">
      <name>fname</name>
      <synopsis>Filter type name</synopsis>
      <typeRef>string[16]</typeRef>
    </component>
    <component componentID="2">
      <name>actions</name>
      <synopsis>The actions graph</synopsis>
      <array type="variable-size">
        <typeRef>filteractinfo</typeRef>
      </array>
    </component>
  </struct>
</dataTypeDef>
```

```
<component componentID="3">
  <name>keys</name>
  <synopsis>Match filter keys</synopsis>
  <optional/>
  <array type="variable-size">
    <typeRef>keyinfo</typeRef>
  </array>
</component>
</struct>
</dataTypeDef>
<!-- For PPPoE LFB -->
<dataTypeDef>
  <name>PPPoETableEntry</name>
  <synopsis>PPPoE Table Entry</synopsis>
  <struct>
    <component componentID="1">
      <name>SubscriberID</name>
      <synopsis>Subscriber Identifier</synopsis>
      <typeRef>uint32</typeRef>
    </component>
    <component componentID="2">
      <name>PPPSessionID</name>
      <synopsis>Session ID</synopsis>
      <typeRef>uint16</typeRef>
    </component>
    <component componentID="3">
      <name>SubscriberMACAddress</name>
      <synopsis>MAC Address</synopsis>
      <typeRef>octetstring[6]</typeRef>
    </component>
    <component componentID="4">
      <name>LocalMACAddress</name>
      <synopsis>Local MAC Address</synopsis>
      <typeRef>octetstring[6]</typeRef>
    </component>
    <component componentID="5">
      <name>MSS</name>
      <synopsis>Maximum Segment Size</synopsis>
      <optional/>
      <typeRef>uint16</typeRef>
    </component>
    <component componentID="6">
      <name>MRU</name>
      <synopsis>Maximum Receive Unit</synopsis>
      <optional/>
      <typeRef>uint16</typeRef>
    </component>
    <component componentID="7">
```

```
<name>MagicNumber</name>
<synopsis>PPP magic number</synopsis>
<optional/>
<typeRef>uint32</typeRef>
</component>
<component componentID="8">
  <name>PeerMagicNumber</name>
  <synopsis>Peer PPP magic number</synopsis>
  <optional/>
  <typeRef>uint32</typeRef>
</component>
</struct>
</dataTypeDef>
<dataTypeDef>
  <name>SubscriberStats</name>
  <synopsis>Subscriber Statistics</synopsis>
  <struct>
    <component componentID="1">
      <name>SubscriberID</name>
      <synopsis>ID of the subscriber</synopsis>
      <typeRef>uint32</typeRef>
    </component>
    <component componentID="2">
      <name>rx_encap_packets</name>
      <synopsis>Total packets received at the encap side</synopsis>
      <typeRef>uint64</typeRef>
    </component>
    <component componentID="3">
      <name>tx_encap_packets</name>
      <synopsis>Total packets transmitted at the encap side</synopsis>
      <typeRef>uint64</typeRef>
    </component>
    <component componentID="4">
      <name>rx_encap_bytes</name>
      <synopsis>Total bytes received at the encap side</synopsis>
      <typeRef>uint64</typeRef>
    </component>
    <component componentID="5">
      <name>tx_encap_bytes</name>
      <synopsis>Total bytes transmitted at the encap side</synopsis>
      <typeRef>uint64</typeRef>
    </component>
    <component componentID="6">
      <name>rx_decap_packets</name>
      <synopsis>Total packets received at the encap side</synopsis>
      <typeRef>uint64</typeRef>
    </component>
    <component componentID="7">
```

```

        <name>tx_decap_packets</name>
        <synopsis>Total packets transmitted at the encap side</synopsis>
        <typeRef>uint64</typeRef>
    </component>
    <component componentID="8">
        <name>rx_decap_bytes</name>
        <synopsis>Total bytes received at the encap side</synopsis>
        <typeRef>uint64</typeRef>
    </component>
    <component componentID="9">
        <name>tx_decap_bytes</name>
        <synopsis>Total bytes transmitted at the encap side</synopsis>
        <typeRef>uint64</typeRef>
    </component>
</struct>
</dataTypeDef>
<!-- For IPv4 Routing LFB -->
<dataTypeDef>
    <name>SubscriberRoutingTableEntry</name>
    <synopsis>A routing table entry</synopsis>
    <struct>
        <component componentID="1">
            <name>SubscriberID</name>
            <synopsis>Subscriber Identifier. Has been generated
            upstream</synopsis>
            <optional/>
            <typeRef>uint32</typeRef>
        </component>
        <component componentID="2">
            <name>IPv4Address</name>
            <synopsis>The destination IPv4 address</synopsis>
            <typeRef>octetstring[4]</typeRef>
        </component>
        <component componentID="3">
            <name>Prefixlen</name>
            <synopsis>The prefix length</synopsis>
            <atomic>
                <baseType>uchar</baseType>
                <rangeRestriction>
                    <allowedRange min="0" max="32"/>
                </rangeRestriction>
            </atomic>
        </component>
        <component componentID="4">
            <name>HopSelector</name>
            <synopsis>
                The HopSelector produced by the prefix matching LFB,
                which will be used as an array index to find next-hop
            </synopsis>
        </component>
    </struct>
</dataTypeDef>

```

```

        information.</synopsis>
        <typeRef>uint32</typeRef>
    </component>
</struct>
</dataTypeDef>
<dataTypeDef>
    <name>IPv4NextHopInfoType</name>
    <synopsis>
        Data type for entry of IPv4 next-hop information table
        in IPv4NextHop LFB. The table uses a hop selector
        received from upstream LFB as a search key to look up
        index of the table to find the next-hop information.
    </synopsis>
    <struct>
        <component componentID="1">
            <name>OutIfiIndex</name>
            <synopsis>
                The interface index of the port that is to pass
                onto downstream LFB, indicating what port this packet
                should be sent out from.</synopsis>
            <typeRef>uint32</typeRef>
        </component>
        <component componentID="2">
            <name>MTU</name>
            <synopsis>
                Maximum Transmission Unit for outgoing port
            </synopsis>
            <typeRef>uint32</typeRef>
        </component>
        <component componentID="3">
            <name>NextHopIPAddr</name>
            <synopsis>The next-hop IPv4 address</synopsis>
            <typeRef>octetstring[4]</typeRef>
        </component>
        <component componentID="4">
            <name>NextHopMACAddr</name>
            <synopsis>The next-hop MAC address</synopsis>
            <typeRef>octetstring[6]</typeRef>
        </component>
    </struct>
</dataTypeDef>
<!-- For PolicerLFB -->
<dataTypeDef>
    <name>PolicerTableEntry</name>
    <synopsis>A routing table entry</synopsis>
    <struct>
        <component componentID="1">
            <name>SubscriberID</name>

```



```
        <synopsis>Subscriber Identifier. Has been generated
        upstream</synopsis>
        <typeRef>uint32</typeRef>
    </component>
    <component componentID="2">
        <name>CIR</name>
        <synopsis>Committed Information Rate</synopsis>
        <typeRef>uint32</typeRef>
    </component>
    <component componentID="3">
        <name>PIR</name>
        <synopsis>Peak Information Rate </synopsis>
        <typeRef>uint32</typeRef>
    </component>
    <component componentID="4">
        <name>CBS</name>
        <synopsis>Committed Burst Size</synopsis>
        <typeRef>uint32</typeRef>
    </component>
    <component componentID="5">
        <name>PBS</name>
        <synopsis>Peak Burst Size</synopsis>
        <typeRef>uint32</typeRef>
    </component>
</struct>
</dataTypeDef>
</dataTypeDefs>
<metadataDefs>
    <metadataDef>
        <name>SubID</name>
        <synopsis>The ID of the subscriber</synopsis>
        <metadataID>1001</metadataID>
        <typeRef>uint32</typeRef>
    </metadataDef>
    <metadataDef>
        <name>OutIfIndex</name>
        <synopsis>Interface Index to output packets</synopsis>
        <metadataID>1002</metadataID>
        <typeRef>uint32</typeRef>
    </metadataDef>
</metadataDefs>
<LFBClassDefs>
    <LFBClassDef LFBClassID="2001">
        <name>Port</name>
        <synopsis>A Port LFB</synopsis>
        <version>1.0</version>
        <inputPorts>
            <inputPort>
```

```

    <name>IngressInPort</name>
    <synopsis>Ingress port from outside the BNG to be
      sent inside</synopsis>
    <expectation>
      <frameExpected>
        <ref>Arbitraty</ref>
      </frameExpected>
    </expectation>
  </inputPort>
  <inputPort>
    <name>EgressInPort</name>
    <synopsis>Egress port from within the BNG to be
      sent outside</synopsis>
    <expectation>
      <frameExpected>
        <ref>Arbitraty</ref>
      </frameExpected>
      <metadataExpected>
        <ref>OutIfIndex</ref>
      </metadataExpected>
    </expectation>
  </inputPort>
</inputPorts>
<outputPorts>
  <outputPort>
    <name>IngressOutPort</name>
    <synopsis>Ingress port to send packets within the
      BNG</synopsis>
    <product>
      <frameProduced>
        <ref>Arbitrary</ref>
      </frameProduced>
    </product>
  </outputPort>
  <outputPort>
    <name>EgressOutPort</name>
    <synopsis>Egress port to send packets out from the
      BNG</synopsis>
    <product>
      <frameProduced>
        <ref>Arbitrary</ref>
      </frameProduced>
    </product>
  </outputPort>
</outputPorts>
<components>
  <component componentID="1" access="read-write">
    <name>ports</name>

```

```

    <synopsis>the table of all ports</synopsis>
    <array type="variable-size">
      <typeRef>PortInfo</typeRef>
    </array>
  </component>
</components>
<events baseID="161">
  <event eventID="1">
    <name>PortChanged</name>
    <synopsis>
      An existing port has been updated.
      When the change occurs we report the table row that has
      changed including its contents + index (port ifindex).
    </synopsis>
    <eventTarget>
      <eventField>ports</eventField>
    </eventTarget>
    <eventChanged/>
    <eventReports>
      <eventReport>
        <eventField>ports</eventField>
        <eventSubscript>_pifindex_</eventSubscript>
      </eventReport>
    </eventReports>
  </event>
  <event eventID="2">
    <name>PortDeleted</name>
    <synopsis>
      An existing port has been deleted.
      When the change occurs we report the table row that
      has changed including its contents + index (port ifindex).
    </synopsis>
    <eventTarget>
      <eventField>ports</eventField>
    </eventTarget>
    <eventDeleted/>
    <eventReports>
      <eventReport>
        <eventField>ports</eventField>
        <eventSubscript>_pifindex_</eventSubscript>
      </eventReport>
    </eventReports>
  </event>
  <event eventID="3">
    <name>PortCreated</name>
    <synopsis>
      A new port has been created. When the change occurs we
      report the table row that has changed including its

```

```

contents + index (port ifindex).
</synopsis>
  <eventTarget>
    <eventField>ports</eventField>
  </eventTarget>
  <eventCreated/>
  <eventReports>
    <eventReport>
      <eventField>ports</eventField>
      <eventSubscript>_pifindex_</eventSubscript>
    </eventReport>
  </eventReports>
</event>
<event eventID="4">
  <name>PortStatsChanged</name>
  <synopsis>
Event used to advertise synchronously port stats. The
ForCES eventInterval property is useful for specifying the
synchronous interval.
</synopsis>
  <eventTarget>
    <eventField>ports</eventField>
  </eventTarget>
  <eventChanged/>
  <eventReports>
    <eventReport>
      <eventField>ports</eventField>
      <eventSubscript>_pifindex_</eventSubscript>
    </eventReport>
  </eventReports>
</event>
</events>
</LFBClassDef>
<LFBClassDef LFBClassID="2002">
  <name>Classifier</name>
  <synopsis>A Classifier LFB. Classifies frames</synopsis>
  <version>1.0</version>
  <inputPorts>
    <inputPort>
      <name>InPort</name>
      <synopsis>Input for the Classifier. Input could be from
Port or tunneling infrastructure</synopsis>
      <expectation>
        <frameExpected>
          <ref>Arbitrary</ref>
        </frameExpected>
      </expectation>
    </inputPort>
  </inputPorts>

```

```
</inputPorts>
<outputPorts>
  <outputPort>
    <name>ControlOut</name>
    <synopsis>Redirects packet towards the control plane.
    </synopsis>
    <product>
      <frameProduced>
        <ref>Arbitrary</ref>
      </frameProduced>
    </product>
  </outputPort>
  <outputPort>
    <name>EthernetOut</name>
    <synopsis>Port to send Ethenet frames</synopsis>
    <product>
      <frameProduced>
        <ref>EthernetFrame</ref>
      </frameProduced>
      <metadataProduced>
        <ref>SubID</ref>
      </metadataProduced>
    </product>
  </outputPort>
  <outputPort>
    <name>IPv4Out</name>
    <synopsis>Port to send IPv4 packets</synopsis>
    <product>
      <frameProduced>
        <ref>IPv4Packet</ref>
      </frameProduced>
      <metadataProduced>
        <ref>SubID</ref>
      </metadataProduced>
    </product>
  </outputPort>
</outputPorts>
<components>
  <component componentID="1" access="read-write">
    <name>Filters</name>
    <synopsis>The table of filters</synopsis>
    <array type="variable-size">
      <typeRef>ClassifierFilterInfo</typeRef>
    </array>
  </component>
</components>
<events baseID="61">
  <event eventID="1">
```

```
<name>FilterChanged</name>
<synopsis>
  A Filter instance has been updated. When the change occurs
  we report the table row that has changed including
  its contents + index.
</synopsis>
<eventTarget>
  <eventField>Filters</eventField>
</eventTarget>
<eventChanged/>
<eventReports>
  <eventReport>
    <eventField>Filters</eventField>
    <eventSubscript>_findex_</eventSubscript>
  </eventReport>
</eventReports>
</event>
<event eventID="2">
  <name>FilterDeleted</name>
  <synopsis>
    An existing Filter has been deleted. When the change
    occurs we report the table row that has changed including
    its contents + index.
  </synopsis>
  <eventTarget>
    <eventField>Filters</eventField>
  </eventTarget>
  <eventDeleted/>
  <eventReports>
    <eventReport>
      <eventField>Filters</eventField>
      <eventSubscript>_findex_</eventSubscript>
    </eventReport>
  </eventReports>
</event>
<event eventID="3">
  <name>FilterCreated</name>
  <synopsis>
    A new filter has been created. When the change occurs
    we report the table row that has changed including
    its contents + index.
  </synopsis>
  <eventTarget>
    <eventField>Filters</eventField>
  </eventTarget>
  <eventCreated/>
  <eventReports>
    <eventReport>
```

```
        <eventField>Filters</eventField>
        <eventSubscript>_findex_</eventSubscript>
    </eventReport>
</eventReports>
</event>
</events>
</LFBClassDef>
<LFBClassDef LFBClassID="2003">
    <name>PPPoE</name>
    <synopsis>PPPoE LFB to encap and decap packets.</synopsis>
    <version>1.0</version>
    <inputPorts>
        <inputPort>
            <name>EncapIn</name>
            <synopsis>A port to encapsulate an IP packet</synopsis>
            <expectation>
                <frameExpected>
                    <ref>IPv4Packet</ref>
                </frameExpected>
                <metadataExpected>
                    <ref>SubID</ref>
                    <ref>OutIfIndex</ref>
                </metadataExpected>
            </expectation>
        </inputPort>
        <inputPort>
            <name>DecapIn</name>
            <synopsis>A port to decapsulate a PPPoE packet</synopsis>
            <expectation>
                <frameExpected>
                    <ref>EthernetFrame</ref>
                </frameExpected>
                <metadataExpected>
                    <ref>SubID</ref>
                </metadataExpected>
            </expectation>
        </inputPort>
    </inputPorts>
    <outputPorts>
        <outputPort>
            <name>EncapOut</name>
            <synopsis>After Encaping an IPv4 Packet create an Ethernet
                frame</synopsis>
            <product>
                <frameProduced>
                    <ref>EthernetFrame</ref>
                </frameProduced>
            </product>
        </outputPort>
    </outputPorts>
</LFBClassDef>
```

```

</outputPort>
<outputPort>
  <name>DecapOut</name>
  <synopsis>Generates IPv4 packets</synopsis>
  <product>
    <frameProduced>
      <ref>IPv4Packet</ref>
    </frameProduced>
    <metadataProduced>
      <ref>SubID</ref>
    </metadataProduced>
  </product>
</outputPort>
</outputPorts>
<components>
  <component componentID="1" access="read-write">
    <name>PPPoEInfo</name>
    <synopsis>Table with PPPoE Subscriber Information</synopsis>
    <array>
      <typeRef>PPPoETableEntry</typeRef>
    </array>
  </component>
  <component componentID="2" access="read-only">
    <name>Stats</name>
    <synopsis>Table with statistics for Encap and Decap
      packets per subscriber</synopsis>
    <array>
      <typeRef>SubscriberStats</typeRef>
    </array>
  </component>
</components>
<events baseID="161">
  <event eventID="1">
    <name>SubChanged</name>
    <synopsis>
      An existing PPPoE Subscriber has been updated.
      When the change occurs we report the table row that has
      changed including its contents + subscriberID.
    </synopsis>
    <eventTarget>
      <eventField>PPPoEInfo</eventField>
    </eventTarget>
    <eventChanged/>
    <eventReports>
      <eventReport>
        <eventField>PPPoEInfo</eventField>
        <eventSubscript>_SubscriberID_</eventSubscript>
      </eventReport>
    </eventReports>
  </event>
</events>

```



```

    </eventReports>
  </event>
  <event eventID="2">
    <name>SubDeleted</name>
    <synopsis>
      An existing PPPoE Subscriber has been deleted.
      When the change occurs we report the table row that has
      changed including its contents + subscriberID.
    </synopsis>
    <eventTarget>
      <eventField>PPPoEInfo</eventField>
    </eventTarget>
    <eventDeleted/>
    <eventReports>
      <eventReport>
        <eventField>PPPoEInfo</eventField>
        <eventSubscript>_SubscriberID_</eventSubscript>
      </eventReport>
    </eventReports>
  </event>
  <event eventID="3">
    <name>SubCreated</name>
    <synopsis>
      A new PPPoE Subscriber has been created.
      When the change occurs we report the table row that has
      changed including its contents + subscriberID.
    </synopsis>
    <eventTarget>
      <eventField>PPPoEInfo</eventField>
    </eventTarget>
    <eventCreated/>
    <eventReports>
      <eventReport>
        <eventField>PPPoEInfo</eventField>
        <eventSubscript>_SubscriberID_</eventSubscript>
      </eventReport>
    </eventReports>
  </event>
  <event eventID="4">
    <name>StatsChanged</name>
    <synopsis>
      Event used to advertise synchronously encap stats. The
      ForCES eventInterval property is useful for specifying the
      synchronous interval.
    </synopsis>
    <eventTarget>
      <eventField>Stats</eventField>
    </eventTarget>

```

```
<eventChanged/>
<eventReports>
  <eventReport>
    <eventField>Stats</eventField>
    <eventSubscript>_SubscriberID_</eventSubscript>
  </eventReport>
</eventReports>
</event>
</events>
</LFBClassDef>
<LFBClassDef LFBClassID="2004">
  <name>IPv4Routing</name>
  <synopsis>IPv4 Routing LFB</synopsis>
  <version>1.0</version>
  <inputPorts>
    <inputPort>
      <name>InPort</name>
      <synopsis>Input port for packets</synopsis>
      <expectation>
        <frameExpected>
          <ref>IPv4Packet</ref>
        </frameExpected>
        <metadataExpected>
          <ref>SubID</ref>
        </metadataExpected>
      </expectation>
    </inputPort>
  </inputPorts>
  <outputPorts>
    <outputPort>
      <name>NormalOut</name>
      <synopsis>Output port for packets</synopsis>
      <product>
        <frameProduced>
          <ref>IPv4Packet</ref>
        </frameProduced>
        <metadataProduced>
          <ref>SubID</ref>
          <ref>OutIfIndex</ref>
        </metadataProduced>
      </product>
    </outputPort>
    <outputPort>
      <name>ExceptionOut</name>
      <synopsis>Port for errors</synopsis>
      <product>
        <frameProduced>
          <ref>IPv4Packet</ref>
        </frameProduced>
      </product>
    </outputPort>
  </outputPorts>
</LFBClassDef>
```

```
        </frameProduced>
    </product>
</outputPort>
</outputPorts>
<components>
  <component componentID="1" access="read-write">
    <name>IPv4RoutingTable</name>
    <synopsis>
      A table for IPv4 Longest Prefix Match. The
      destination IPv4 address of every input packet is
      used as a search key to look up the table to find
      out a next-hop selector.
    </synopsis>
    <array>
      <typeRef>SubscriberRoutingTableEntry</typeRef>
    </array>
  </component>
  <component componentID="2" access="read-write">
    <name>IPv4NextHopTable</name>
    <synopsis>
      The IPv4NextHopTable component. A
      HopSelector is used to match the table index
      to find out a row that contains the next-hop
      information result.
    </synopsis>
    <array>
      <typeRef>IPv4NextHopInfoType</typeRef>
    </array>
  </component>
</components>
</LFBClassDef>
<LFBClassDef LFBClassID="2005">
  <name>Policer</name>
  <synopsis>Policer LFB</synopsis>
  <version>1.0</version>
  <inputPorts>
    <inputPort>
      <name>InUpstreamPort</name>
      <synopsis>Input port for the Policer LFB for upstream
        traffic</synopsis>
      <expectation>
        <frameExpected>
          <ref>IPv4Packet</ref>
        </frameExpected>
        <metadataExpected>
          <ref>SubID</ref>
        </metadataExpected>
      </expectation>
    </inputPort>
  </inputPorts>

```

```
</inputPort>
<inputPort>
  <name>InDownstreamPort</name>
  <synopsis>Input port for the Policer LFB for downstream
    traffic</synopsis>
  <expectation>
    <frameExpected>
      <ref>IPv4Packet</ref>
    </frameExpected>
    <metadataExpected>
      <ref>SubID</ref>
    </metadataExpected>
  </expectation>
</inputPort>
</inputPorts>
<outputPorts>
  <outputPort>
    <name>OutUpstreamPort</name>
    <synopsis>Output port for the Policer LFB for upstream
      traffic</synopsis>
    <product>
      <frameProduced>
        <ref>IPv4Packet</ref>
      </frameProduced>
      <metadataProduced>
        <ref>SubID</ref>
      </metadataProduced>
    </product>
  </outputPort>
  <outputPort>
    <name>OutDownstreamPort</name>
    <synopsis>Output port for the Policer LFB for downstream
      traffic</synopsis>
    <product>
      <frameProduced>
        <ref>IPv4Packet</ref>
      </frameProduced>
      <metadataProduced>
        <ref>SubID</ref>
      </metadataProduced>
    </product>
  </outputPort>
</outputPorts>
<components>
  <component componentID="1" access="read-write">
    <name>UpstreamPolicy</name>
    <synopsis>Policy entries for upstream traffic</synopsis>
    <array>
```

```

        <typeRef>BandwidthTableEntry</typeRef>
      </array>
    </component>
    <component componentID="2" access="read-write">
      <name>DownstreamPolicy</name>
      <synopsis>Policy entries for downstream traffic</synopsis>
      <array>
        <typeRef>PolicerTableEntry</typeRef>
      </array>
    </component>
  </components>
</LFBClassDef>
</LFBClassDefs>
</LFBLibrary>

```

Figure 9: BNG XML model

## 8. Acknowledgements

The activities of Evangelos Haleplidis have been carried out with funding provided via the StandICT.eu initiative, funded with Grant Agreement no. 780439 under the European Commission's Horizon 2020 Programme.

## 9. IANA Considerations

TBD

## 10. Security Considerations

TBD

## 11. References

### 11.1. Normative References

- [I-D.haleplidis-bcause-forces-gap-analysis]  
Haleplidis, E. and J. Salim, "ForCES architecture CUSP applicability", draft-haleplidis-bcause-forces-gap-analysis-00 (work in progress), April 2019.
- [I-D.ietf-quic-transport]  
Iyengar, J. and M. Thomson, "QUIC: A UDP-Based Multiplexed and Secure Transport", draft-ietf-quic-transport-23 (work in progress), September 2019.

- [RFC2516] Mamakos, L., Lidl, K., Evarts, J., Carrel, D., Simone, D., and R. Wheeler, "A Method for Transmitting PPP Over Ethernet (PPPoE)", RFC 2516, DOI 10.17487/RFC2516, February 1999, <<https://www.rfc-editor.org/info/rfc2516>>.
- [RFC3746] Yang, L., Dantu, R., Anderson, T., and R. Gopal, "Forwarding and Control Element Separation (ForCES) Framework", RFC 3746, DOI 10.17487/RFC3746, April 2004, <<https://www.rfc-editor.org/info/rfc3746>>.
- [RFC5810] Doria, A., Ed., Hadi Salim, J., Ed., Haas, R., Ed., Khosravi, H., Ed., Wang, W., Ed., Dong, L., Gopal, R., and J. Halpern, "Forwarding and Control Element Separation (ForCES) Protocol Specification", RFC 5810, DOI 10.17487/RFC5810, March 2010, <<https://www.rfc-editor.org/info/rfc5810>>.
- [RFC5811] Hadi Salim, J. and K. Ogawa, "SCTP-Based Transport Mapping Layer (TML) for the Forwarding and Control Element Separation (ForCES) Protocol", RFC 5811, DOI 10.17487/RFC5811, March 2010, <<https://www.rfc-editor.org/info/rfc5811>>.
- [RFC5812] Halpern, J. and J. Hadi Salim, "Forwarding and Control Element Separation (ForCES) Forwarding Element Model", RFC 5812, DOI 10.17487/RFC5812, March 2010, <<https://www.rfc-editor.org/info/rfc5812>>.
- [RFC7121] Ogawa, K., Wang, W., Haleplidis, E., and J. Hadi Salim, "High Availability within a Forwarding and Control Element Separation (ForCES) Network Element", RFC 7121, DOI 10.17487/RFC7121, February 2014, <<https://www.rfc-editor.org/info/rfc7121>>.
- [TR-101] Broadband Forum, "TR-101 - Migration to Ethernet-Based Broadband Aggregation", 2011, <[https://www.broadband-forum.org/download/TR-101\\_Issue-2.pdf](https://www.broadband-forum.org/download/TR-101_Issue-2.pdf)>.

## 11.2. Informative References

- [media1] "Forces-vzn", , 06 2016, <<https://www.sdxcentral.com/articles/news/verizon-uses-radisys-mojatatu-sdn-nfv/2016/06/>>.
- [media2] "Forces-vzn2", , 06 2016, <<https://www.sdxcentral.com/articles/news/meet-mojatatu-quiet-company-verizon-chose-sdn/2016/06/>>.

- [media3] "Forces-Large-Scale-Kubernetes", , 10 2019,  
<<https://www.cengn.ca/mojatatu-analysis-kubernetes-deployments/>>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate  
Requirement Levels", BCP 14, RFC 2119,  
DOI 10.17487/RFC2119, March 1997,  
<<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC6956] Wang, W., Haleplidis, E., Ogawa, K., Li, C., and J.  
Halpern, "Forwarding and Control Element Separation  
(ForCES) Logical Function Block (LFB) Library", RFC 6956,  
DOI 10.17487/RFC6956, June 2013,  
<<https://www.rfc-editor.org/info/rfc6956>>.

## Authors' Addresses

Evangelos Haleplidis  
M. Aleksandrou 29B  
Paiania, Athens 19002  
Greece

Email: [ehalep@gmail.com](mailto:ehalep@gmail.com)

Jamal Hadi Salim  
Mojatatu Networks  
Suite 200, 15 Fitzgerald Road  
Ottawa, Ontario K2H 9G1  
Canada

Email: [hadi@mojatatu.com](mailto:hadi@mojatatu.com)

Jae Won Chung  
Viasat  
6155 El Camino Real  
Carlsbad 92009  
USA

Email: [JaeWon.Chung@viasat.com](mailto:JaeWon.Chung@viasat.com)

rtgwg  
Internet-Draft  
Intended status: Informational  
Expires: May 6, 2020

S. Homma  
NTT  
X. de Foy  
InterDigital Inc.  
A. Galis  
University College London  
LM. Contreras  
Telefonica  
November 3, 2019

Gateway Function for Network Slicing  
draft-homma-rtgwg-slice-gateway-01

Abstract

This document describes the roles and requirements for a slice gateway that is a function or function group for handling data plane traffic, such as connecting/disconnecting and compose/decompose network slice subnet instances and providing network slices from end to end. The interworkings between management and control elements at the management and control planes with the gateway function for controlling and orchestrating end-to-end network slices are also presented in this document.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

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."

This Internet-Draft will expire on May 6, 2020.

Copyright Notice

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 (<https://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.

## Table of Contents

|  |    |
|--|----|
| 1. Introduction . . . . .  | 3  |
| 2. Definition of Terms . . . . .   | 4  |
| 3. Motivations and Roles of SLG . . . . .                                      | 6  |
| 4. Architecture of Network Slicing System . . . . .                            | 8  |
| 4.1. Network Slice Management System Architecture . . . . .                    | 8  |
| 5. Requirements for SLG . . . . .  | 10 |
| 5.1. Management of NS as Infrastructure . . . . .                              | 10 |
| 5.1.1. Data Plane Aspect . . . . .   | 10 |
| 5.1.1.1. Identification/Classification . . . . .                               | 11 |
| 5.1.1.2. Transporting/Forwarding . . . . .                                     | 11 |
| 5.1.1.3. Isolation among NSs . . . . .   | 12 |
| 5.1.1.4. Service Chaining as Infrastructural<br>Mechanism(*Optional) . . . . . | 13 |
| 5.1.2. Control/Management Planes Aspects . . . . .                             | 13 |
| 5.1.2.1. Interfaces to Controllers or Operation Systems . . . . .              | 13 |
| 5.1.2.2. Address Resolution/Routing . . . . .                                  | 13 |
| 5.1.2.3. Authentication Authorization Accounting (AAA) . . . . .               | 13 |
| 5.1.2.4. Operation Administration and Maintenance(OAM) . . . . .               | 14 |
| 5.1.2.5. Traffic Monitoring . . . . .  | 14 |
| 5.2. Management of Services on NS (*Optional) . . . . .                        | 14 |
| 5.2.1. Data Plane Aspect . . . . .   | 14 |
| 5.2.1.1. Identification/Classification . . . . .                               | 14 |
| 5.2.1.2. QoS Control . . . . .   | 14 |
| 5.2.1.3. Steering/Service Chaining(Cooperation with VNFs) . . . . .            | 15 |
| 5.2.2. Control/Management Planes Aspects . . . . .                             | 15 |
| 5.2.2.1. Interfaces to Service Management Systems . . . . .                    | 15 |
| 5.2.2.2. Collection of Telemetry information . . . . .                         | 15 |
| 6. Structure of SLG . . . . .  | 15 |
| 7. Deployment of SLG . . . . .   | 16 |
| 7.1. Examples of Components Required to Maintain SLG Functions . . . . .       | 16 |
| 7.2. SLG Types Depending on Locations on NS . . . . .                          | 17 |
| 7.2.1. Edge SLG(E-SLG) . . . . .   | 17 |
| 7.2.2. Inter-Subnet SLG(IS-SLG) . . . . .                                      | 17 |
| 7.2.3. Inter-Domain SLG(ID-SLG) . . . . .                                      | 17 |
| 7.3. Horizontal Connection . . . . .   | 17 |
| 7.4. Vertical Connection . . . . .   | 20 |

|  |    |
|--|----|
| 7.5. Software vs. Hardware . . . . .                             | 21 |
| 8. Interconnection between NSSIs . . . . .                       | 21 |
| 8.1. Pre-arrangement of transport protocols . . . . .            | 21 |
| 8.2. Quality Assurance between SLGs . . . . .                    | 21 |
| 8.3. Secure Interconnection . . . . .                            | 22 |
| 9. Interfaces of SLG Controller . . . . .                        | 22 |
| 9.1. Southbound Interface . . . . .                              | 22 |
| 9.2. Northbound Interface for Higher Operation Systems . . . . . | 22 |
| 9.3. Northbound Interface for Customers/Tenants . . . . .        | 22 |
| 10. Security Considerations . . . . .                            | 22 |
| 11. IANA Considerations . . . . .                                | 22 |
| 12. Acknowledgement . . . . .                                    | 22 |
| 13. Informative References . . . . .                             | 23 |
| Appendix A. Requirements for each SLG Type . . . . .             | 25 |
| Appendix B. Position of SLG on ETSI NFV MANO . . . . .           | 26 |
| Appendix C. Complementation of Network Slicing in 3GPP . . . . . | 27 |
| Authors' Addresses . . . . .                                     | 27 |

## 1. Introduction

Network slicing is an approach to create separate virtual networks in support of service depending on several requirements on the same physical resources, and it enables networks to adapt to requirements, which is diverse more, inexpensively and flexibly. The overview is introduced in [Slicing\_Tutorial] and [NECOS].

It's also expected to enhance usability of infrastructural networks for tenants and create new business opportunities. For example, by using network slices lent from infrastructure operators, other industrial companies can provide communication services including ensurance of network transport without having physical infrastructure.

From a business point of view, a slice includes a combination of all the relevant network resources, functions, and assets required to fulfill a specific business case or service, including OSS, BSS and DevOps processes.

From the network infrastructure point of view, network slice requires the partitioning and assignment of a set of resources that can be used in an isolated, disjunctive or non- disjunctive manner for that slice.

From the tenant point of view, network slice provides different capabilities, specifically in terms of their management and control capabilities, and how much of them the network service provider hands over to the slice tenant. As such there are two kinds of slices: (A) Inner slices, understood as the partitions used for internal services

of the provider, retaining full control and management of them. (B) Outer slices, being those partitions hosting customer services, appearing to the customer as dedicated networks.

Network slices are established with combination of various technologies, such as software defined network (SDN), network function virtualization (NFV), or traffic engineering, and managed/operated with automation technologies such as orchestrator.

Assumed use cases of network slices include establishment of virtual networks whose qualities are guaranteed from end to end under the supervision of multi-domain orchestrators. In such cases, a network slice subnet is created on each domain, such as access network and core network, and an end-to-end network slices is composed of connected subnets.

Network slice subnets are built based on specifications of the underlay network, and thus the used technologies might vary. Therefore, a gateway function, which enables to connect subnets while adapting the differentiations and forward data packets to/from the appropriate next subnet, is required.

This document describes the gateway function for network slicing, called slice gateway or SLG, and its role and requirements. Note that defining a new data plane technology is not a goal of this draft. In addition, this draft aims to specify management-related requirements for an SLG, which may be implemented using existing data plane technologies.

## 2. Definition of Terms

This section describes definitions and terminologies related to network slicing, especially gateway function and interconnection network slices established in each domain. Other complementary definitions are described in [I-D.homma-slice-provision-models].

**Network Slicing:** Network slicing is a technology or an approach to create separate logical networks in support of services, depending on several requirements, on the same physical resources. This is possible by combinations of several network technologies.

**Network Slice (NS):** An NS is a logical separate network that provides specific network capabilities and characteristics.

**Network Slice Instance (NSI):** An NSI is a logical network instance composed with required infrastructure resources, including networking (WAN), computing (NFVI) resources, and some include additional network service functions such as firewall or load-

balancer. It is composed of one or more Network Slice Subnet Instances.

**Network Slice Subnet:** An NS subnet is a representation of a set of resources structuring a part of NSI within a single domain.

**Network Slice Subnet Instance (NSSI):** An NSSI is a partial logical network instance represented as a network slice instance. It is a minimal unit managed or provided as a network slice. One or more NSSI structure an NSI or an E2E-NSI.

**End-to-End Network Slice Instance (E2E-NSI):** An E2E-NSI is an NSI providing connectivity among end points. An E2E-NSI is used for emphasizing end to end connectivity provided by an NSI.

**Network Slice as a Service (NSaaS):** An NSaaS is a service delivery model in which a third-party provider (e.g., vertical customer) hosts NSs and makes them available to customers. In this model, there are mainly two roles: NS provider and NS tenant.

**Network Slice Provider (NS Provider):** An NS provider is a person or group that designs and instantiates one or more NSIs/NSSIs, and provides them to NS tenants. In some cases, an NS provider is an infrastructure operator simultaneously. This includes NSI, NSSI, and E2E-NSI providers.

**Network Slice Tenant (NS Tenant):** An NS tenant is a person or group that rents and occupies NSs from NS providers.

**Domain:** A domain is a group of a network and devices administrated as a unit with common rules and procedures.

**Administrative Domain:** An administrative domain is a group of networks and devices managed by an administrator.

**Resource:** A resource is element used to create virtual networks. There are several types of resources, i.e., connectivity, computing and storage.

**Network Function Virtualization (NFV):** NFV is the concept or technologies to provide dedicated network appliances as software.

**Software Defined Network (SDN):** SDN is the concept or technologies to separate network control plane from data plane, and control network devices dynamically and flexibly.

**Virtual Network:** A virtual network is a network running a number of virtual network functions.

**Virtual Network Function (VNF):** A virtual network function (VNF) is a network function whose functional software is decoupled from hardware. One or more virtual machines running different software and processes on top of industry-standard high-volume servers, switches and storage, or cloud computing infrastructure, and capable of implementing network functions traditionally implemented via custom hardware appliances and middleboxes (e.g., router, NAT, firewall, load balancer, etc.)

**Slice Gateway Function (SLG):** An SLG is a function or a group of functions to connect/disconnect NSSIs. The roles are described in the following sections.

**Business Support System and Operation Support System (BSS/OSS):** BSS/OSS are systems to support service providing and operation of network devices.

**Orchestrator:** Orchestrator is an entity to operate network components automatically. There are several types of orchestrators including NFV Orchestrator (NFVO) or service orchestrator defined by ETSI NFV and Open Source MANO (OSM) ([NFV-Architectural-Framework] and [OSM-White-Paper]).

**SLG Controller (SLG-Ctrl):** An SLG-Ctrl is an entity that controls SLGs. An SLG-Ctrl is controlled by upper-level operation systems such as OSS/BSS or orchestrator.

### 3. Motivations and Roles of SLG

One of the main roles of SLG is the enablement of interworkings between data plane with management and control elements for controlling and orchestrating end-to-end slices.

Use cases of network slices are discussed in several Standard Developing Organizations (SDOs). Some examples are described in use cases document ([I-D.netslices-usecases]).

In some proposed use cases, an NS is structured across multiple network domains. The capability of NSSIs might be different because the components are domain-specific. In particular, the differentiation in capability between different administrative domains is large.

Moreover, several variations can be considered on NS provisioning in NSaaS (ref. [I-D.homma-slice-provision-models]), and some cases need abstraction of underlay infrastructure to NS tenants. SLG solution provides controllability of network functions for manipulation of

NSs intensively, and it can be expected to emphasize the manageability of NSIs in such cases.

For connecting some different NSSIs and providing a NS that guarantees the prescribed quality from end to end, SLGs are required to connect such NSSIs. SLGs enable to provide E2E-NSIs independently of specifications of underlay networks by hiding the differentiations and connecting between NSSIs. An overview of this concept is shown in Figure 1. SLGs glue NSSIs established on each domain and provide an E2E-NSI.

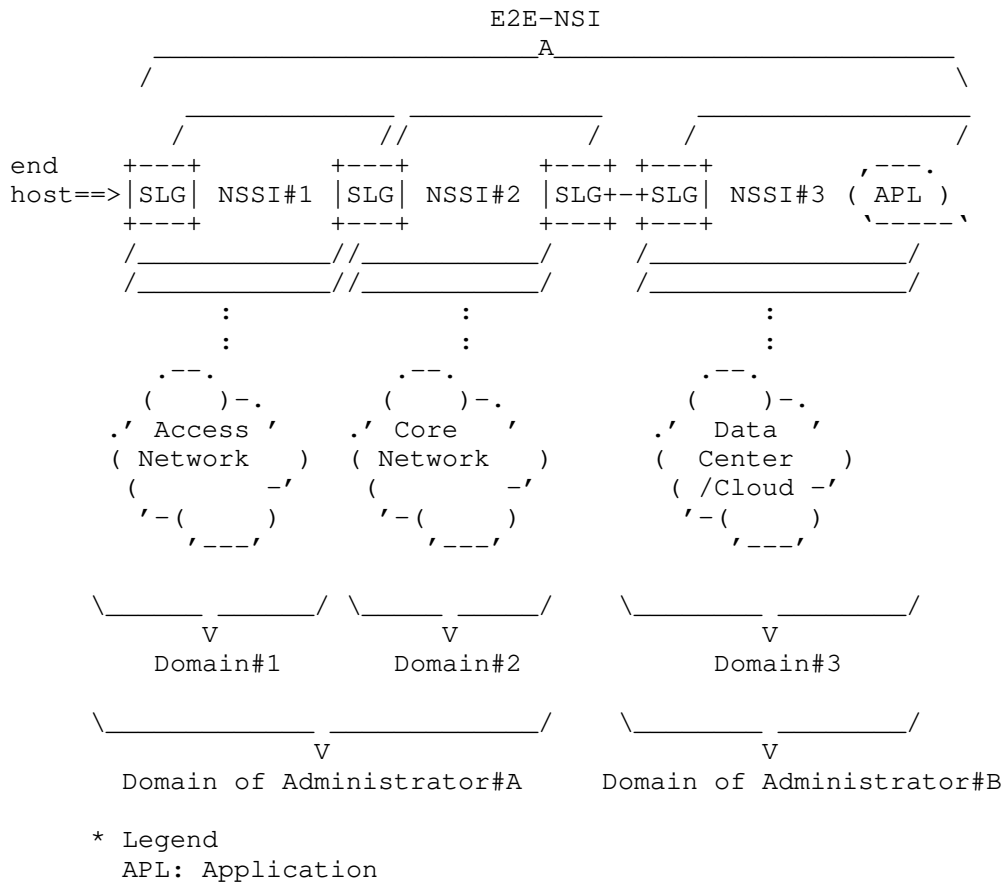


Figure 1: E2E-NSI composed of multiple NSSIs

Moreover, identification of user service traffic and their allocation/disallocation to the appropriate NSSI are required at the

edges of E2E-NSIs, as shown in Figure 2, and SLGs might take on these roles.

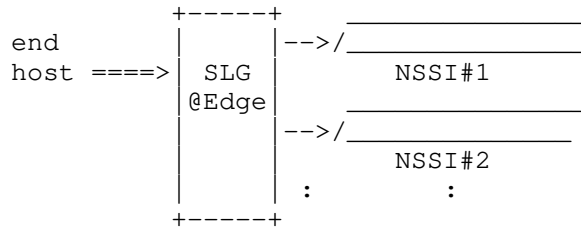


Figure 2: NSSI selection of SLG

Note that, this model has the assumption that transitions of data packets from one NSSI to another are executed at only SLGs. Also, an SLG is not necessarily implemented as a single device or virtual machine (VM).

#### 4. Architecture of Network Slicing System

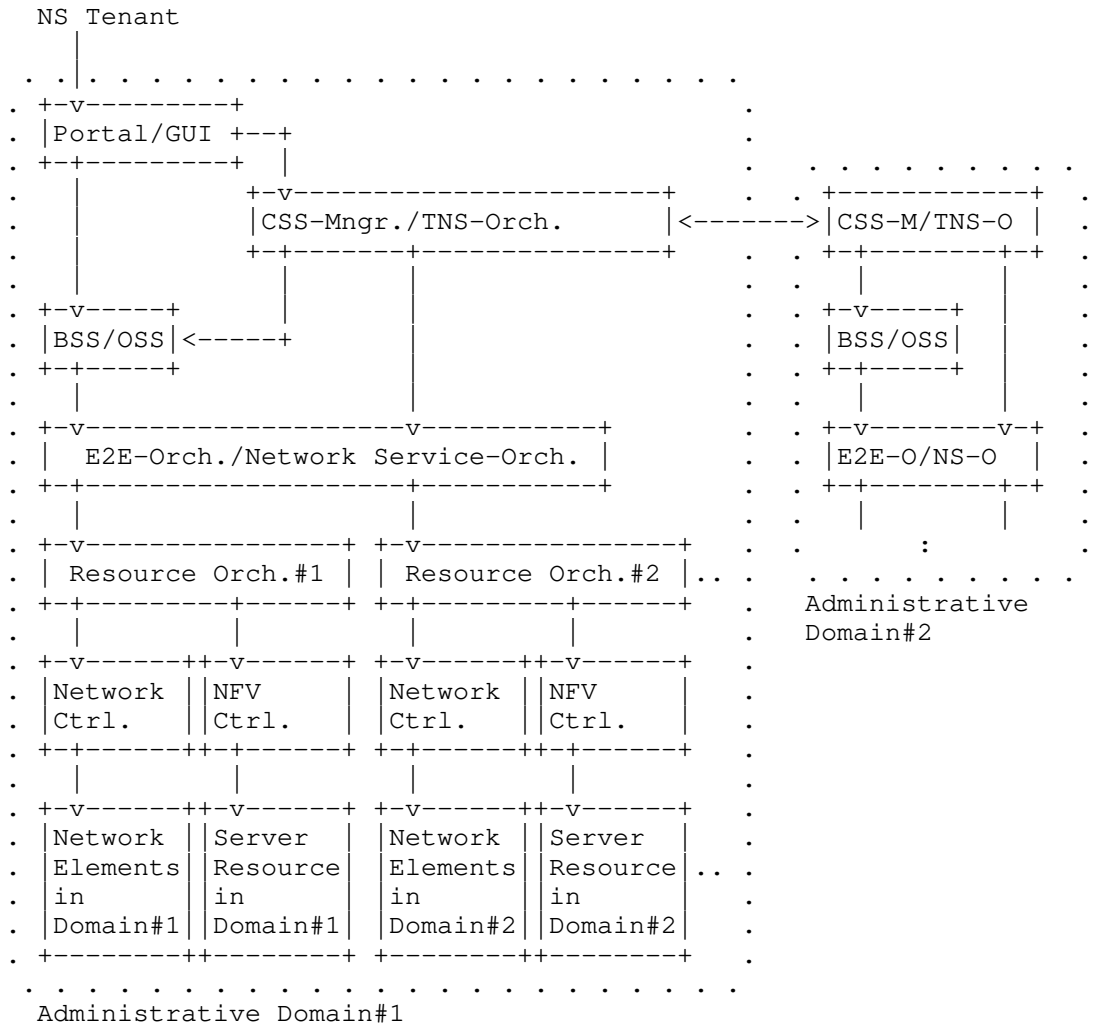
NSs are composed of several (virtual) network functions and links, and the characteristics of each NS are based on the assumed service. Also, some of NSs are deployed accross multiple administrative domains. For deploying the appropriate NSs based on each service requirements, a management system, which enables to control network resources totally within a domain, and interaction between such management systems are required.

An SLG is a network function, and SLGs are installed at edge of NSSIs. NSs are dynamically created, deleted, and moved depending on requests from network opertor orNS tenants. Therefore, some SLGs would be required to be VNF for flexible deployment.

This section describes overview of NS management system architecture (Section 4.1) .

##### 4.1. Network Slice Management System Architecture

The architecture overview of NS management system is shown in Figure 3.



CSS-Mngr./CSS-M:Cross-Segment Slice Manager  
TNS-Orch./TNS-O:Transport Network Slice Orchestrator

Figure 3: Overview of NS Management Architecture

Orchestrators manage whole resources including network elements and server resources (i.e., routing, bandwidth, compute or storage). In this figure, the resources including network elements and server resources are managed by resource orchestrators installed in each domain, and the E2E-orchestrator and network service orchestrator handle resource orchestrators.



NSs are requested from NS tenants via the portal system and the order of creations of an NS is given to the E2E orchestrator from the portal system via BSS/OSS. When an NS across multiple administrative domains are requested, the portal system that received the request forwards the order to create NSSIs to the other infrastructure providers' systems via Cross-Segment Slice Manager. The details of COMS architecture are described in the architecture document ([I-D.qiang-coms-architecture]).

SLGs are also controlled via orchestrators. An SLG basically belongs to a network element, and it might also belong to server resource if it runs as a VNF. (The position of SLG deployed as a VNF is shown in Appendix B.)

The information model used in this architecture is described in information model document ([I-D.qiang-coms-netslicing-information-model]).

## 5. Requirements for SLG

An SLG is basically a component in the data plane and has the roles of data packet processing. Moreover, it is required to have functions for control/management processes such as connecting to underlay networks or managing NSs.

Furthermore, an SLG might be required to support handling services provided on NSs in addition to controlling of NS because an SLG is an edge node on an E2E-NSI.

In this section, we describe the requirements for an SLG in terms of the following aspects and their interworkings.

1. Data plane for NSs as infrastructure
2. Control/management plane for NSs as infrastructure
3. Data plane for services on NSs
4. Control/management plane for services on NSs

### 5.1. Management of NS as Infrastructure

#### 5.1.1. Data Plane Aspect

#### 5.1.1.1. Identification/Classification

SLGs at the edge of E2E-NSs MUST have the capability to identify and classify data packets, and assign them to the appropriate E2E-NS. This requirement varies depending on the location.

**Fixed Access:** An SLG MUST identify and classify data packet with access point, including CPE or WiFi-AP, or subscriber ID such as VLAN-ID. Moreover, in some services, an SLG should identify and classify data packets based on user device or application used in the communication.

**Mobile Access:** An SLG MUST identify and classify data packet with subscriber-ID such as IMSI, radio-wave bandwidth, or identifier of tunnels. Moreover, in some services, an SLG should identify and classify data packets based on application used in the communication or location of the user equipment (UE).

**Connection Point between NSSI:** An SLG MUST identify and classify data packet based on the tunnel-ID or virtual routing and forwarding (VRF) that received the packets. If specific slice identifier such as a value mapped in the metadata field of the IP header is used; an SLG should identify and classify data packets with the ID.

#### 5.1.1.2. Transporting/Forwarding

SLGs MUST provide functions for transport data packets depending on the specifications of the underlay networks.

**Encapsulation/Decapsulation/Tagging:** In network slicing, duplication of IP addresses of user packets between NSs MUST be accepted, thus, using techniques that enable separation of a network logically is preferred. In short, some tunnel protocols or tagging approaches should be used as transport of NSs. For this reason, SLG MUST support encapsulation or tagging of data packets based on the specification of the underlay network. Also, SLG MUST support the packets' decapsulation or untagging. Examples of tunnel protocols and tags that can be used for creating NSs on L2/L3 segments are described below.

**L2 Segment:** VLAN, MPLS, Segment Routing MPLS (SR-MPLS), PPPoE, etc.

**L3 Segment:** GRE, L2TP, GTP-U, VxLAN, IPv6 Segment Routing (SRv6), etc.



Quality of service (QoS) Control: If there is an order of priority between NSs on the same underlay infrastructure, an SLG should remark the appropriate QoS parameter of the outer-most header of each packet following the preconfigured setting and provide packet scheduling based on the QoS parameter for providing priority control. The field that SLG refers may vary depending on the specification of the underlay network. For example, COS value is remarked in L2 segments; on the other hand, DSCP value is remarked in L3 segments.

#### 5.1.1.4. Service Chaining as Infrastructural Mechanism(\*Optional)

If an SLG is composed of a combination of several components, a service chaining mechanism is required to make them work together and achieve SLG functionality.

Moreover, some NSs may traverse NFVs such as firewalls or cache servers for providing value-added services to their users. In such cases, SLG might be required to support service chaining mechanisms, such as handling of network service header (NSH) defined in [RFC8300]. If an NS includes the service chaining architecture defined in [RFC7665], some SLG would be required to support following functions; classifier(CF), service function forwarder (SFF), and inter boundary node(IBN). (Details of CF, SFF and IBN are described in SFC documents; [RFC7665], [RFC8459].)

#### 5.1.2. Control/Management Planes Aspects

##### 5.1.2.1. Interfaces to Controllers or Operation Systems

SLG MUST have interface to its controller or operation systems for set parameters related to the data plane functions described in Section 5.1.1. In addition, an SLG at the edges of E2E-NSs MUST have interfaces to authentication servers.

##### 5.1.2.2. Address Resolution/Routing

An SLG MUST support address resolution or routing mechanisms to connect to underlay network elements including routers or L2 switches.

##### 5.1.2.3. Authentication Authorization Accounting (AAA)

For preventing entry of irregular traffic to NSs, an SLG at the edge of E2E-NS MUST support AAA mechanism for incoming traffic. Also, when an SLG connects to another SLG in other administrative domain, SLGs should have a mechanism to confirm that the connection is established with the regular processes. For example, an SLG is

required to support authentication of the opponent SLG with key information indicated from higher-level operation systems.

#### 5.1.2.4. Operation Administration and Maintenance (OAM)

In management of NSs, OAM mechanisms for both underlay and overlay networks is required for SLGs. For an underlay network, an SLG MUST have OAM functions to confirm connectivity to interconnect equipment. For an overlay network as an NS, an SLG MUST have OAM functions to confirm connectivity to the nodes on the same NS.

#### 5.1.2.5. Traffic Monitoring

An SLG shall support monitoring of traffic amount and latency as a mechanism for checking whether each of the accommodated NSs is satisfying its SLA. When an NS can't fulfill its SLA, the SLG MUST send a notification to any listening system.

### 5.2. Management of Services on NS (\*Optional)

#### 5.2.1. Data Plane Aspect

##### 5.2.1.1. Identification/Classification

In NSaaS, some NS tenants may need delivery of an individual service to each user, device, or application on the same NS. For such service deliveries, an SLG might be required to identify and classify user traffic based on some information such as subscriber ID or payload of data packets. Also, an SLG should be controllable from the NS tenant.

##### 5.2.1.2. QoS Control

An NS accommodates several communication devices and SLGs might be required to have fair queueing mechanisms for maintaining service quality of each user. Also, different types of service traffic that have different priorities might coexist on an NS. For example, some NS providers might provide telephone and internet access services to their users with an NS. In such cases, SLG might be required to provide QoS control mechanisms for enforcing priority control based on service priorities.

These QoS controls are executed depending on the information of inner packets and are independent of isolation mechanisms as infrastructure. An SLG might be required to have a hierarchical QoS control mechanism in case that both QoS controls for services over NSs and isolation between NSs are required.

#### 5.2.1.3. Steering/Service Chaining(Cooperation with VNFs)

SLG might be required to support steering or service chaining function for conveying data packets to the appropriate network functions deployed on an NS based on the classification result and user's contract information.

#### 5.2.2. Control/Management Planes Aspects

##### 5.2.2.1. Interfaces to Service Management Systems

An SLG might have interfaces to controllers for managing user policies on each NS. Some controllers might be deployed on the same NS. If some controllers are located at external networks, they might require SLGs to have APIs.

##### 5.2.2.2. Collection of Telemetry information

In an NSaaS, collection of telemetry information of each NS might be required for understanding traffic usage. Thus, an SLG might be required to support to collect and report telemetry information of connected NSs.

### 6. Structure of SLG

SLG is composed of data plane entities and controller. SLG Data plane entity (SLG-D) has functions to manipulate NSSIs and to handle traffic on slices. In some cases, an SLG-D is composed of several physical devices and/or virtual instances. Function types supported by SLG-D are listed Section 5. SLG controller (SLG-C) accommodates multiple SLG Data plane entities via its southbound interface, and sets configurations into each SLG-D. SLG-C also has a northbound interface and it provides accesses to two endpoints: higher operation systems such as orchestrator, and to external customers and tenants which own their NSSIs. Then, functionality and controllability exposed to customers/tenants should be limited, and the access must be secure (i.e., authentication and admission control should be supported). The overview of SLG structure is shown in Figure 5.

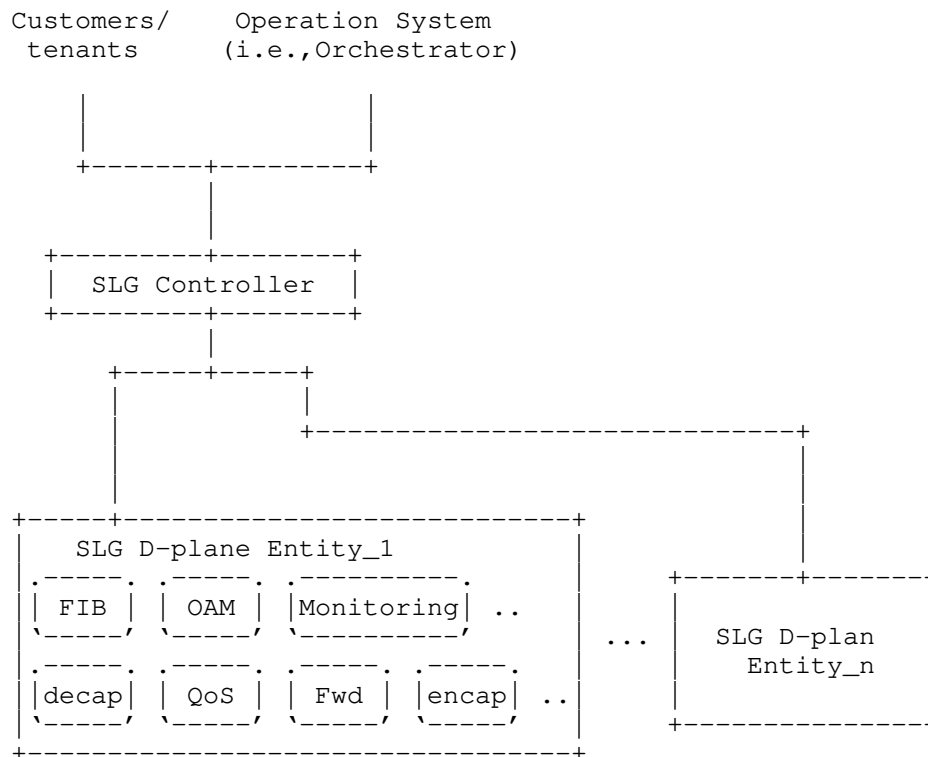


Figure 5: Overview of SLG Structure

## 7. Deployment of SLG

This section describes considerations related with deployment of SLGs.

### 7.1. Examples of Components Required to Maintain SLG Functions

For providing E2E-NSs on existing network infrastructures, some components located at boundaries of domains are required to have the same set of functionality as an SLG. Examples of such components in each domain type are described below.

Fixed Network: CPE/HGW, Service Edge, Gateway Router, etc.

Mobile Network: User Equipment, Radio-AP, eNodeB, S/P-GW ([TS.36.300-3GPP]), etc.

Data Center: Gateway Router, L2 switch, ToR switch, Server, etc.

## 7.2. SLG Types Depending on Locations on NS

There are mainly three types of SLG for creating E2E-NS across multiple administrative domains. The requirements of each SLG type are listed in Appendix A.

### 7.2.1. Edge SLG(E-SLG)

E-SLG is located at an edge of an E2E-NS, and supports identification, classification and authentication of user traffic in addition to fundamental SLG functions, such as transport and isolation. Also, it might be required to have capabilities for services delivered on an NS.

### 7.2.2. Inter-Subnet SLG(IS-SLG)

IS-SLG is located between NSSIs within a single administrative domain and has only fundamental functions such as QoS control or translation of headers.

This type of SLG enables to separate an NSI into some NSSIs. It will provide modularities of NSSIs, and simplify the management of NSIs. However, it is not necessarily required if a common transport mechanism in all domains is used.

### 7.2.3. Inter-Domain SLG(ID-SLG)

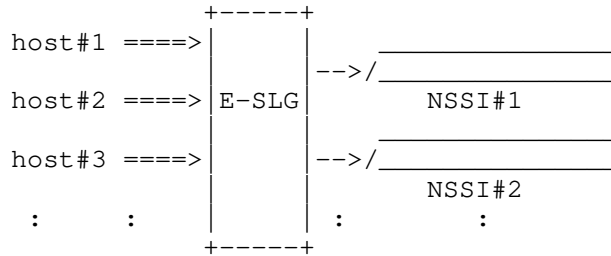
ID-SLG is located between NSSIs established on different domains. It supports authentication for connecting to the opponent SLG in addition to fundamental functions.

## 7.3. Horizontal Connection

The connection form of an SLG varies depending on which type it is. Examples of horizontal connection forms of each SLG type are described below.

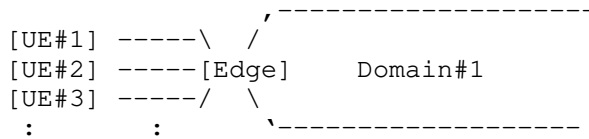
E-SLG: An E-SLG accommodates several hosts and NSSIs. This has a forwarding table of end hosts and insert their packets to the appropriate NSSI. An overview of this connection is shown in Figure 6.



**\*Virtual Layer\***

////////////////////////////////////

**\*Physical Layer\***

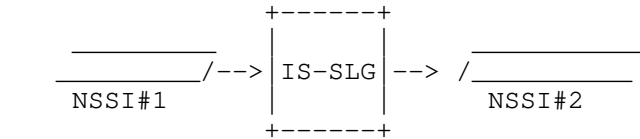


Edge: Edge Node

Figure 6: Overview of horizontal connection of E-SLG

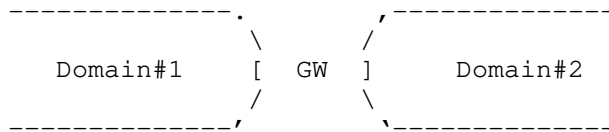
**IS-SLG:** An IS-SLG has the role of mediator between NSSIs and passes packets received from an NSSI to the next one. If transport methods used in each domain are different, the IS-SLG translate packet form to the appropriate one. An overview of this connection is shown in Figure 7.

\*Virtual Layer\*



////////////////////////////////////

\*Physical Layer\*

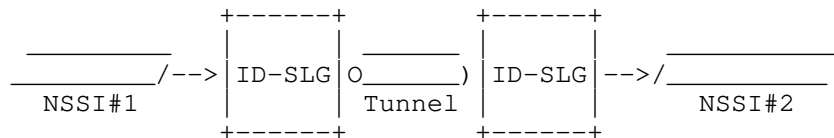


GW: Gateway Node

Figure 7: Overview of horizontal connection of IS-SLG

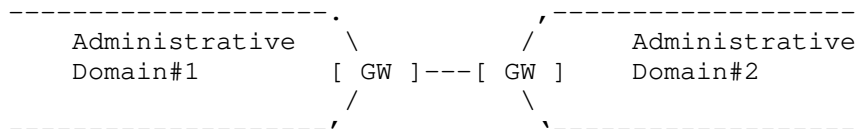
ID-SLG: An ID-SLG passes data packets to another ID-SLG located on a different administrative domain. Some tunnel established between them in advance may be used for the passing of packets. An overview of this connection is shown in Figure 8.

\*Virtual Layer\*



////////////////////////////////////

\*Physical Layer\*



GW: Gateway Node

Figure 8: Overview of horizontal connection of ID-SLG

#### 7.4. Vertical Connection

There are two patterns of vertical connection of SLGs in the middle of E2E-NSs. The first pattern is that the SLGs accommodate only a set of NSSIs, which are composition of the same E2E-NS. In this pattern, such SLGs are not required to support NSSI selection, however, establishment of a new SLG is required when a new E2E-NS is created. This might causes extra overheads because of deploying many SLGs.

The other pattern is that such SLGs are acceptable to accommodate multiple NSSIs from each domain. In this pattern, SLGs support NSSI selection. On the other hand, this pattern can restrain the number of SLGs. Also, it is easy to provide transit of data packets from an NSSI to another NSSI on the same domain.

The overviews of these patterns are shown in Figure 9 and Figure 10.

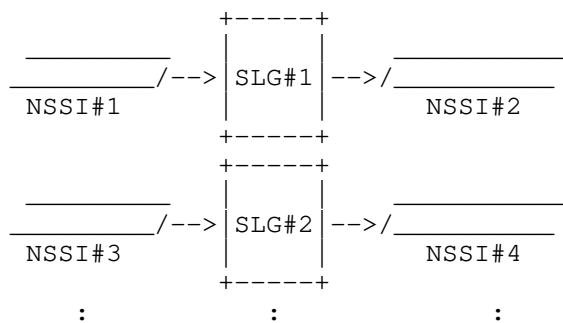


Figure 9: Overview of vertical connection of SLG: Separated Pattern

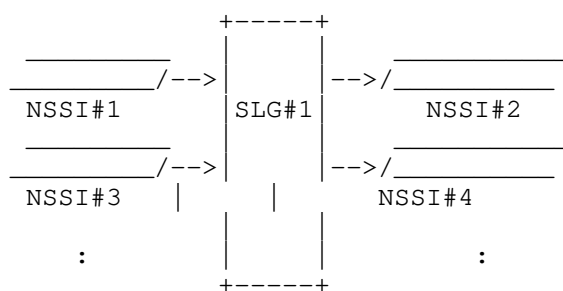


Figure 10: Overview of vertical connection of SLG: Shared Pattern

### 7.5. Software vs. Hardware

An SLG can be created as either a software or hardware function. NSs are virtual networks created depending on requests from external NS tenants, and thus software would be more compatible with usage for NSs in terms of flexibility or manageability. Moreover, it enables to increase or decrease for each function if SLG is composed of combination of several components. However, it is difficult to provide high performance or sufficient throughput for carrier-grade networks with software function. In addition, it would be difficult to implement sufficient QoS control mechanisms with general servers, because they requires special hardware structures. An example of position of SLG in NFV environment is described in Appendix B.

On the other hand, hardware appliances are able to provide high throughput compared with software. However, they are inflexible in terms of provisioning.

From the above considerations, operators should prepare SLG in appropriate ways depending on their usages or locations.

## 8. Interconnection between NSSIs

SLG provides interconnectivity between NSSIs. The concept and fundamental framework including the related NS information model are described in NSSIs interconnection document ([I-D.defoy-coms-subnet-interconnection]).

This section is focused on interconnection between NSSIs established on different administrative domains, and describes considerations related to this condition.

### 8.1. Pre-arrangement of transport protocols

For interconnection between different administrative NSSIs, pre-arrangement of the transport protocol, which is used to connect between SLGs is required. Orchestration systems indicate the protocol and configuration to each SLG.

### 8.2. Quality Assurance between SLGs

In addition to establishing connection, quality control of communication is important. SLGs of egress side should execute traffic shaping to prevent some NSs from excessively occupying the link between SLGs. Moreover, some SLGs are connected to several other SLGs that are deployed on the different locations. Therefore SLGs of the ingress side should execute traffic policing to avoid

excessive inflow of traffic into some NSs. The parameters for these controls are pre-configured by orchestration systems.

The above approaches are ones of the simplest ways to provide quality assurance of inter-administrative subnets. If there is stricter isolation request, more considerations would be required.

### 8.3. Secure Interconnection

For connecting networks of different administrators, secure interconnection schemes are required. Especially, in an NSaaS, networks might be connected to several networks, and schemes for ensuring secure connectivity would be more important.

SLGs confirm whether the opponent SLG is regular when it requests to connect, and reject the request if the SLG is not regular. In some cases, SLGs might be confirm whether the inner packets received from the other SLGs are sent from regular users.

## 9. Interfaces of SLG Controller

### 9.1. Southbound Interface

SLG-C supports protocols to communicate with SLG-Ds. Information and parameters exchanged between SLG-D and SLG-C are TBD.

### 9.2. Northbound Interface for Higher Operation Systems

TBD

### 9.3. Northbound Interface for Customers/Tenants

TBD

## 10. Security Considerations

Requirements and considerations for SLG related to security are described in Section 5 and Section 8.

## 11. IANA Considerations

This memo includes no request to IANA.

## 12. Acknowledgement

The authors would like to thank Li Qiang for her kind review and valuable feedback.

## 13. Informative References

- [I-D.defoy-coms-subnet-interconnection]  
Foy, X., Rahman, A., Galis, A., kiran.makhijani@huawei.com, k., and L. Qiang, "Interconnecting (or Stitching) Network Slice Subnets", draft-defoy-coms-subnet-interconnection-01 (work in progress), October 2017.
- [I-D.homma-slice-provision-models]  
Homma, S., Nishihara, H., Miyasaka, T., Galis, A., OV, V., Lopez, D., Contreras, L., Ordonez-Lucena, J., Martinez-Julia, P., Qiang, L., Rokui, R., Ciavaglia, L., and X. Foy, "Network Slice Provision Models", draft-homma-slice-provision-models-00 (work in progress), February 2019.
- [I-D.ietf-6man-segment-routing-header]  
Previdi, S., Filsfils, C., Raza, K., Dukes, D., Leddy, J., Field, B., daniel.voyer@bell.ca, d., daniel.bernier@bell.ca, d., Matsushima, S., Leung, I., Linkova, J., Aries, E., Kosugi, T., Vyncke, E., Lebrun, D., Steinberg, D., and R. Raszuk, "IPv6 Segment Routing Header (SRH)", draft-ietf-6man-segment-routing-header-08 (work in progress), January 2018.
- [I-D.ietf-spring-segment-routing-mpls]  
Filsfils, C., Previdi, S., Bashandy, A., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing with MPLS data plane", draft-ietf-spring-segment-routing-mpls-11 (work in progress), October 2017.
- [I-D.netslices-usecases]  
kiran.makhijani@huawei.com, k., Qin, J., Ravindran, R., Geng, L., Qiang, L., Peng, S., Foy, X., Rahman, A., Galis, A., and G. Fioccola, "Network Slicing Use Cases: Network Customization and Differentiated Services", draft-netslices-usecases-02 (work in progress), October 2017.
- [I-D.qiang-coms-netslicing-information-model]  
Qiang, L., Galis, A., 67, 4., kiran.makhijani@huawei.com, k., Martinez-Julia, P., Flinck, H., and X. Foy, "Technology Independent Information Model for Network Slicing", draft-qiang-coms-netslicing-information-model-01 (work in progress), October 2017.

- [I-D.rokui-5g-transport-slice]  
Rokui, R., Homma, S., Lopez, D., Foy, X., Contreras, L.,  
Ordonez-Lucena, J., Martinez-Julia, P., Boucadair, M.,  
Eardley, P., Makhijani, K., and H. Flinck, "5G Transport  
Slice Connectivity Interface", draft-rokui-5g-transport-  
slice-00 (work in progress), July 2019.
- [NECOS] NECOS, "Novel Enablers for Cloud Slicing",  
<<http://www.h2020-necos.eu>>.
- [NFV-Architectural-Framework]  
Network Functions Virtualisation (NFV) ETSI Industry  
Specification Group (ISG), "Network Functions  
Virtualisation (NFV); Architectural Framework", December  
2014, <[http://www.etsi.org/deliver/etsi\\_gs/  
NFV/001\\_099/002/01.02.01\\_60/gs\\_NFV002v010201p.pdf](http://www.etsi.org/deliver/etsi_gs/NFV/001_099/002/01.02.01_60/gs_NFV002v010201p.pdf)>.
- [OSM-White-Paper]  
ETSI, "OSM White Paper", October 2016,  
<[https://osm.etsi.org/images/OSM-Whitepaper-TechContent-  
ReleaseONE-FINAL.pdf](https://osm.etsi.org/images/OSM-Whitepaper-TechContent-ReleaseONE-FINAL.pdf)>.
- [RFC7348] Mahalingam, M., Dutt, D., Duda, K., Agarwal, P., Kreeger,  
L., Sridhar, T., Bursell, M., and C. Wright, "Virtual  
eXtensible Local Area Network (VXLAN): A Framework for  
Overlaying Virtualized Layer 2 Networks over Layer 3  
Networks", RFC 7348, DOI 10.17487/RFC7348, August 2014,  
<<https://www.rfc-editor.org/info/rfc7348>>.
- [RFC7665] Halpern, J., Ed. and C. Pignataro, Ed., "Service Function  
Chaining (SFC) Architecture", RFC 7665,  
DOI 10.17487/RFC7665, October 2015,  
<<https://www.rfc-editor.org/info/rfc7665>>.
- [RFC8300] Quinn, P., Ed., Elzur, U., Ed., and C. Pignataro, Ed.,  
"Network Service Header (NSH)", RFC 8300,  
DOI 10.17487/RFC8300, January 2018,  
<<https://www.rfc-editor.org/info/rfc8300>>.
- [RFC8459] Dolson, D., Homma, S., Lopez, D., and M. Boucadair,  
"Hierarchical Service Function Chaining (hSFC)", RFC 8459,  
DOI 10.17487/RFC8459, September 2018,  
<<https://www.rfc-editor.org/info/rfc8459>>.

## [Slicing\_Tutorial]

IEEE NetSoft2018, "Network Slicing Landscape Tutorial",  
June 2018,  
<<http://netsoft2018.ieee-netsoft.org/program/tutorials/>;  
<http://discovery.ucl.ac.uk/10051374/>>.

## [TS.23.501-3GPP]

3rd Generation Partnership Project (3GPP), "3GPP TS 23.501  
(V16.0.0): System Architecture for 5G System; Stage 2",  
September 2018, <[http://www.3gpp.org/ftp//Specs/  
archive/23\\_series/23.501/23501-g00.zip](http://www.3gpp.org/ftp//Specs/archive/23_series/23.501/23501-g00.zip)>.

## [TS.36.300-3GPP]

3rd Generation Partnership Project (3GPP), "Evolved  
Universal Terrestrial Radio Access (E-UTRA) and Evolved  
Universal Terrestrial Radio Access Network (E-UTRAN);  
Overall description; Stage 2", December 2007,  
<<http://www.qtc.jp/3GPP/Specs/36300-830.pdf>>.

## Appendix A. Requirements for each SLG Type

The requirements for each SLG type are listed in Figure 11.

|   | E-SLG | IS-SLG | ID-SLG | Reference        |
|---|-------|--------|--------|------------------|
| *Data-Plane of NS as Infrastructure               |       |        |        |                  |
| Identification/<br>Classification                 | M     | O      | O      | Section 5.1.1.1. |
| Transport/<br>Forwarding                          | M     | O      | M      | Section 5.1.1.2. |
| Isolation   | M     | M      | M      | Section 5.1.1.3. |
| Service Chain                                     | O     | O      | O      | Section 5.1.1.4. |
| *Control/Management-Plane of NS as Infrastructure |       |        |        |                  |
| IF to Ctrl/OpS                                    | M     | M      | M      | Section 5.1.2.1. |
| Addr Resolution<br>/Routing                       | M     | M      | M      | Section 5.1.2.2. |
| AAA   | M     | -      | M      | Section 5.1.2.3. |
| OAM   | M     | M      | M      | Section 5.1.2.4. |



|   |   |   |   |                  |
|---|---|---|---|------------------|
| Monitoring                                  | M | M | M | Section 5.1.2.5. |
| *Data-Plane for Service on NS               |   |   |   |                  |
| Identification/<br>Classification           | O | - | O | Section 5.2.1.1. |
| QoS Control                                 | O | O | O | Section 5.2.1.2. |
| Steering/<br>Service Chain                  | O | - | O | Section 5.2.1.3. |
| *Control/Management-Plane for Service on NS |   |   |   |                  |
| IF to Service<br>Manager                    | O | O | O | Section 5.2.2.1. |
| Telemetry                                   | O | O | O | Section 5.2.2.2. |

M: Mandatry, O: Optional, - : Not Required

Figure 11: List of Requirements for each SLG

## Appendix B. Position of SLG on ETSI NFV MANO

Some SLGs and the controllers are deployed and run on NSs as VNFs. An arechitecture for managing lifecycle of VNFs is under standardization in ETS NFV MANO.

The mapping of SLG as a VM into ETSI NFV MANO architecture is described in Figure 12. In some cases, SLGs are deployed with container. VNFs are parts of NS compositions and NFV orchestrator would be controlled by upper control entities such as resource orchestrator.

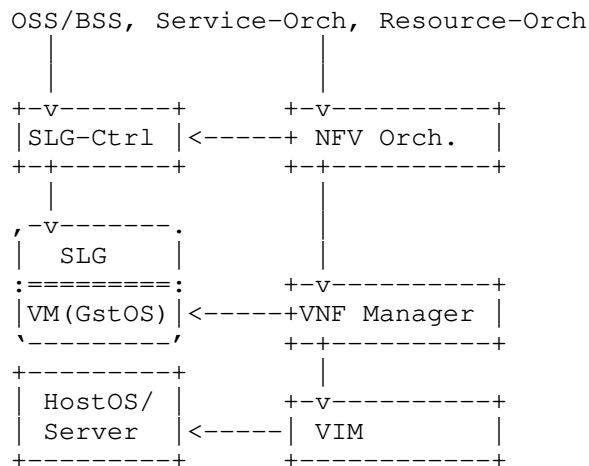


Figure 12: Position of SLG as a VM on ETSI NFV MANO

#### Appendix C. Complementation of Network Slicing in 3GPP

The 3GPP 5GS is natively support network slicing (ref. [TS.23.501-3GPP], and UPF provides some functions for manipulation of NSs, such as NS selection, QoS control, traffic steering, etc. 3GPP is responsible for standardizing user plane manipulation for mobility management, and interworking with transport on underlay network and external networks of 5GS such as DNS is out of scope in 3GPP.

SLG will provide complementary definitions of functions and interfaces for providing E2E-NSI including 5GS. A way of interworking between transport NS and RAN/UPF is described in [I-D.rokui-5g-transport-slice].

Further study is TBD.

#### Authors' Addresses

Shunsuke Homma  
NTT  
Japan

Email: shunsuke.homma.fp@hco.ntt.co.jp

Xavier de Foy  
InterDigital Inc.  
Canada

Email: Xavier.Defoy@InterDigital.com

Alex Galis  
University College London  
United Kingdom

Email: a.galis@ucl.ac.uk

Luis M. Contreras  
Telefonica  
Ronda de la Comunicacion, s/n  
Sur-3 building, 3rd floor  
Madrid 28050  
Spain

Email: luismiguel.contrerasmurillo@telefonica.com  
URI: <http://lmcontreras.com/>

Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: 13 October 2022

Z. Hu  
Huawei Technologies  
H. Chen  
Futurewei  
J. Yao  
Huawei Technologies  
C. Bowers  
Juniper Networks  
Y. Zhu  
China Telecom  
Y. Liu  
China Mobile  
11 April 2022

SR-TE Path Midpoint Restoration  
draft-hu-spring-segment-routing-proxy-forwarding-19

Abstract

Segment Routing Traffic Engineering (SR-TE) supports explicit paths using segment lists containing adjacency-SIDs, node-SIDs and binding-SIDs. The current SR FRR such as TI-LFA provides fast re-route protection for the failure of a node along a SR-TE path by the direct neighbor or say point of local repair (PLR) to the failure. However, once the IGP converges, the SR FRR is no longer sufficient to forward traffic of the path around the failure, since the non-neighbors of the failure will no longer have a route to the failed node. This document describes a mechanism for the restoration of the routes to the failure of a SR-MPLS TE path after the IGP converges. It provides the restoration of the routes to an adjacency segment, a node segment and a binding segment of the path. With the restoration of the routes to the failure, the traffic is continuously sent to the neighbor of the failure after the IGP converges. The neighbor as a PLR fast re-routes the traffic around the failure.

Requirements Language

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

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

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."

This Internet-Draft will expire on 13 October 2022.

#### Copyright Notice

Copyright (c) 2022 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 (<https://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 Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

#### Table of Contents

|   |    |
|---|----|
| 1. Introduction . . . . .                                       | 3  |
| 1.1. Terminology . . . . .                                      | 3  |
| 2. Proxy Forwarding . . . . .                                   | 4  |
| 3. Protocol Extensions/Re-uses for Proxy Forwarding . . . . .   | 4  |
| 3.1. Advertising Binding Segment . . . . .                      | 4  |
| 3.2. Advertising Proxy Forwarding . . . . .                     | 5  |
| 4. Proxy Forwarding Example . . . . .                           | 6  |
| 4.1. Advertising Proxy Forwarding . . . . .                     | 8  |
| 4.2. Building Proxy Forwarding Table . . . . .                  | 8  |
| 4.3. Proxy Forwarding for Binding Segment . . . . .             | 9  |
| 5. Security Considerations . . . . .                            | 10 |
| 6. Acknowledgements . . . . .                                   | 10 |
| 7. References . . . . .   | 10 |
| 7.1. Normative References . . . . .                             | 10 |
| 7.2. Informative References . . . . .                           | 11 |
| Appendix A. Proxy Forwarding for Adjacency and Node Segment . . | 11 |
| A.1. Next Segment is an Adjacency Segment . . . . .             | 11 |
| A.2. Next Segment is a Node Segment . . . . .                   | 12 |
| Authors' Addresses . . . . .                                    | 13 |

## 1. Introduction

Segment Routing Traffic Engineering (SR-TE) is a technology that implements traffic engineering using a segment list. SR-TE supports the creation of explicit paths using adjacency-SIDs, node-SIDs, anycast-SIDs, and binding-SIDs. A node-SID in the segment list defining an SR-TE path indicates a loose hop that the SR-TE path should pass through. When the node fails, the network may no longer be able to properly forward traffic on that SR-TE path.

[I-D.ietf-rtgwg-segment-routing-ti-lfa] describes an SR FRR mechanism that provides fast re-route protection for the failure of a node on a SR-TE path by the direct neighbor or say point of local repair (PLR) to the failure. However, once the IGP converges, the SR FRR is no longer sufficient to forward traffic of the path around the failure, since the non-neighbors of the failure will no longer have a route to the failed node and drop the traffic.

To solve this problem,

[I-D.ietf-spring-segment-protection-sr-te-paths] proposes that a hold timer should be configured on every router in a network. After the IGP converges on the event of a node failure, if the node-SID of the failed node becomes unreachable, the forwarding changes should not be communicated to the forwarding planes on all configured routers (including PLRs for the failed node) until the hold timer expires. This solution may not work for some cases such as some of nodes in the network not supporting this solution.

This document describes a proxy forwarding mechanism for the restoration of the routes to the failure of a SR-MPLS TE path after the IGP converges. It provides the restoration of the routes to an adjacency segment, a node segment and a binding segment on a failed node along the path. With the restoration of the routes to the failure, the traffic for the SR-MPLS TE path is continuously sent to the neighbor of the failure after the IGP converges. The neighbor as a PLR fast re-routes the traffic around the failure.

### 1.1. Terminology

SR: Segment Routing.

PLR: Point of Local Repair.

LSP: Link State Protocol Data Unit (PDU) in IS-IS.

LSA: Link State Advertisement in OSPF.

LS: Link State, which is LSP or LSA.

## 2. Proxy Forwarding

In the proxy forwarding mechanism, each neighbor of a possible failed node advertises its SR proxy forwarding capability in its network domain when it has the capability. This capability indicates that the neighbor (the Proxy Forwarder) will forward traffic on behalf of the failed node. A router receiving the SR Proxy Forwarding capability from the neighbors of a failed node will send traffic using the node-SID of the failed node to the nearest Proxy Forwarder after the IGP converges on the event of the failure.

Once the affected traffic reaches a Proxy Forwarder, it sends the traffic on the post-failure shortest path to the node immediately following the failed node in the segment list.

For a binding segment of a possible failed node, the node advertises the information about the binding segment, including the binding SID and the list of SIDs/segments associated with the binding SID, to its direct neighbors only. Note that the information is not advertised in the network domain.

After the node fails and the IGP converges on the failure, the traffic with the binding SID of the failed node will reach its neighbor having SR Proxy Forwarding capability. Once receiving the traffic, the neighbor swaps the binding SID with the list of SIDs/segments associated with the binding SID and sends the traffic along the post-failure shortest path to the first node in the segment list.

## 3. Protocol Extensions/Re-uses for Proxy Forwarding

This section describes the semantic of protocol extensions/re-uses for advertising the information about each binding segment (including its binding SID and the list of SIDs/segments associated with the binding SID) of a node to its direct neighbors and the SR proxy forwarding capability of a node in a network domain.

### 3.1. Advertising Binding Segment

For a binding segment (or binding for short) on a node A, which consists of a binding SID and a list of SIDs/segments, node A advertises an LS containing the binding (i.e., the binding SID and the list of the SIDs/segments) in a binding segment TLV. The LS is advertised only to each of the node A's neighboring nodes. For OSPFv2, the LS is a opaque LSA of LS type 9 (i.e., a link local scope LSA). For IS-IS, the TLV is advertised in Circuit Scoped Link State PDUs (CS-LSP) [RFC7356].

Alternatively, when a protocol (such as PCE or BGP running on a controller) supports sending a binding on a node A to A, this protocol may be extended to send the binding with node A to A's neighbors if the controller knows the neighbors and there are protocol (PCE or BGP) sessions between the controller and the neighbors.

Note: how to send bindings of node A to A's neighbors via which protocol is out of the scope of this document.

### 3.2. Advertising Proxy Forwarding

When a node P is able to do SR proxy forwarding for its neighboring nodes for protecting the failures of these nodes, P advertises its SR proxy forwarding capability for these nodes. The mirror SID [RFC8402] for a node N (Neighbor of P) advertised by P using IS-IS extensions [RFC8667] indicates the capability of P for N.

For a node X in the network, it learns the prefix/node SID of node N, which is originated and advertised by node N. It creates a proxy prefix/node SID of node N for node P if node P is capable of doing SR proxy forwarding for node N. The proxy prefix/node SID of node N for node P is a copy of the prefix/node SID of node N originated by node N, but stored under (or say, associated with) node P. The route to the proxy prefix/node SID is through proxy forwarding capable nodes.

In normal operations, node X prefers to use the prefix/node SID of node N. When node N fails, node X prefers to use the proxy prefix/node SID of node N. Thus node X will forward the traffic targeting to the prefix/node SID of node N to node P when node N fails, and node P will do a SR proxy forwarding for node N and forward the traffic towards its final destination without going through node N.

Note that the behaviors of normal IP forwarding and routing convergences in a network are not changed at all by the SR proxy forwarding. For example, the next hop used by BGP is an IP address (or prefix). The IGP and BGP converge in normal ways for changes in the network. The packet with its IP destination to this next hop is forwarded according to the IP forwarding table (FIB) derived from IGP and BGP routes.

Similar to IS-IS [RFC8667], OSPF should be extended for advertising mirror SID to indicate the capability. Note that OSPF extensions is out of the scope of this document.



#### 4. Proxy Forwarding Example

This section illustrates the proxy forwarding for a binding SID through an example. The proxy forwarding for a node SID and an adjacency SID can refer to [I-D.ietf-spring-segment-protection-sr-te-paths] or Appendix. Figure 1 is an example network topology used to illustrate the proxy forwarding mechanism for a binding SID. Each node N has SRGB = [N000-N999]. RT1 is an ingress node of SR domain. RT3 is a failure node. RT2 is a Point of Local Repair (PLR) node, i.e., a proxy forwarding node. Label Stack 1 uses a node-SID and a binding SID. The Binding-SID with label=100 at RT3 represents the ECMP-aware path RT3->RT4->RT5. So Label Stack 1, which consists of the node-SID for RT3 following by Binding-SID=100, represents the ECMP-aware path RT1->RT3->RT4->RT5.

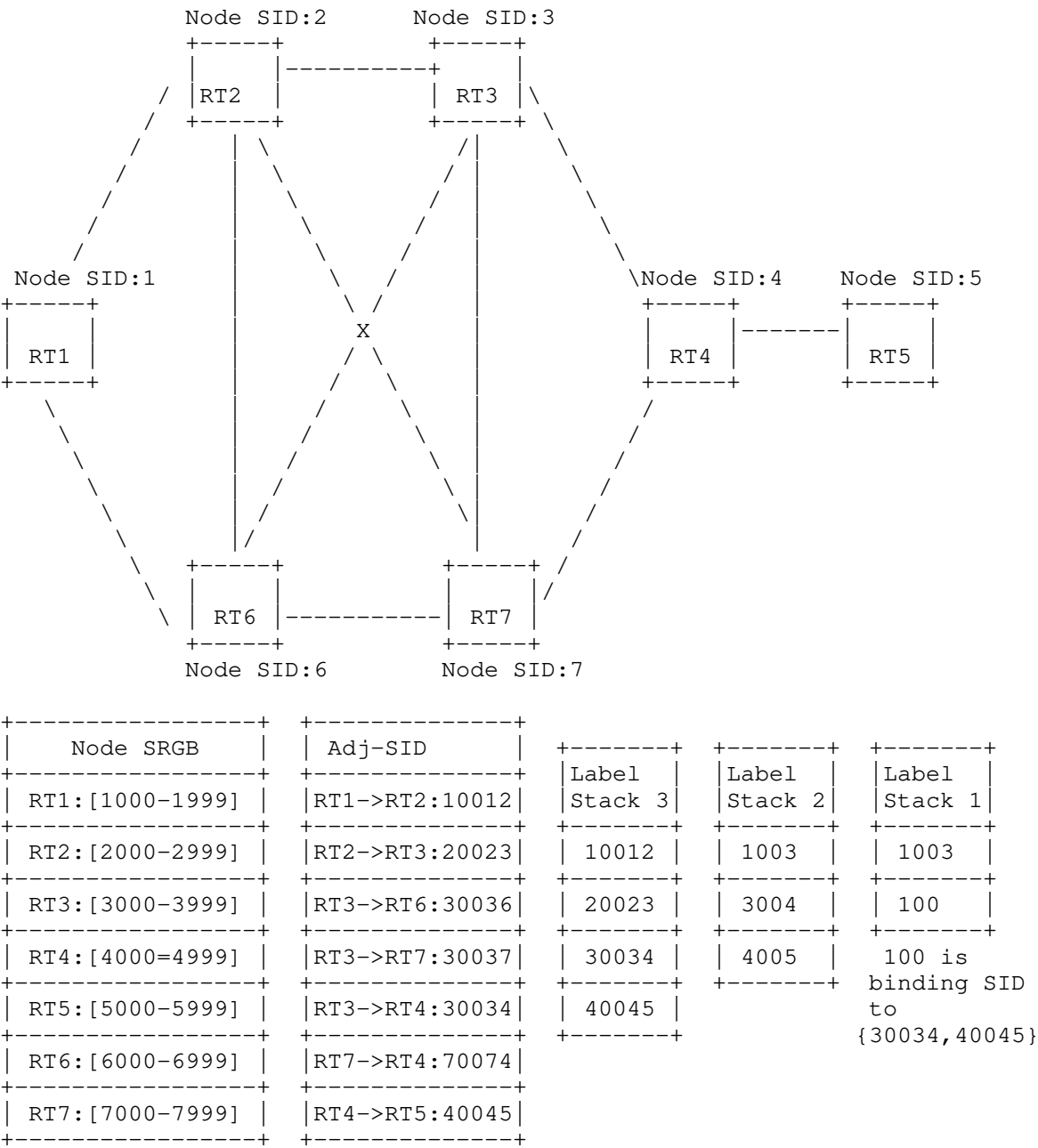


Figure 1: Topology of SR-TE Path

#### 4.1. Advertising Proxy Forwarding

If the Point of Local Repair (PLR), for example, RT2, has the capability to do SR proxy forwarding for its neighboring nodes such as RT3, it must advertise this capability. When RT3 fails, RT2 needs to maintain its SR proxy forwarding capability for a period of time. When the proxy forwarding table corresponding to the fault node is deleted, the capability is withdrawn. The nodes in the network (for example, RT1) learn the prefix/node SID advertised by RT3 and the proxy forwarding capability for RT3 advertised by RT2. When RT3 is normal, the nodes prefer prefix/node SID. When the RT3 fails, the proxy prefix/node SIDs of RT3 for RT2 is preferred.

For binding-SID 100, which is associated with segment list {30034, 40045}, RT3 advertises the binding (i.e., 100 bond to {30034, 40045}) to its neighbors RT2, RT4 and RT7. RT2 as PLR uses the binding to build an entry for proxy forwarding for binding-SID 100 in its Proxy Forwarding Table for RT3. The entry is used when RT3 fails.

#### 4.2. Building Proxy Forwarding Table

A SR proxy node P needs to build an independent proxy forwarding table for each neighbor N. The proxy forwarding table for node N contains the following information:

- 1: Node N's SRGB range and the difference between the SRGB start value of node P and that of node N;
- 2: Every adjacency-SID of N and Node-SID of the node pointed to by node N's adjacency-SID.
- 3: Every binding-SID of N and the label stack associated with the binding-SID.

Node P (PLR) uses a proxy forwarding table based on the next segment to find a node N as a backup forwarding entry to the adjacency-SID and Node-SID of node N. When node N fails, the proxy forwarding table needs to be maintained for a period of time, which is recommended for 30 minutes.

Node RT3 in Figure 1 is node N, and node RT2 is node P (PLR). RT2 builds the proxy forwarding table for RT3. RT2 calculates the proxy forwarding table for RT3, as shown in Figure 2.

| In-label | SRGBDiffValue | Next Label | Action                   | Map Label |
|----------|---------------|------------|--------------------------|-----------|
| 2003     | -1000         | 30034      | Fwd to RT4               | 2004      |
|          |               | 30036      | Fwd to RT6               | 2006      |
|          |               | 30037      | Fwd to RT7               | 2007      |
|          |               | 100        | Swap to { 30034, 40045 } |           |

Figure 2: RT2's Proxy Forwarding Table for RT3

#### 4.3. Proxy Forwarding for Binding Segment

This Section shows through example how a proxy node uses the SR proxy forwarding mechanism to forward traffic to the destination node when a node fails and the next segment of label stack is a binding-SID.

As shown in Figure 1, Label Stack 1 {1003, 100} represents SR-TE loose path RT1->RT3->RT4->RT5, where 100 is a Binding-SID, which represents segment list {30034, 40045}.

When the node RT3 fails, the proxy forwarding SID implied or advertised by the RT2 is preferred to forward the traffic of the RT1 to the PLR node RT2. Node RT2 acts as a PLR node and uses Binding-SID to query the proxy forwarding table locally built for RT3. The path returned is the label forwarding path to RT3's next hop node (RT4), which bypasses RT3. The specific steps are as follows:

- a. RT1 swaps label 1003 to out-label 2003 to RT3.
- b. RT2 receives the label forwarding packet whose top label of label stack is 2003, and searches for the local Routing Table, the behavior found is to lookup Proxy Forwarding table due to RT3 failure.
- c. RT2 uses Binding-SID:100 (label 2003 has pop) as the in-label to lookup the Next Label record of the Proxy Forwarding Table, the behavior found is to swap to Segment list {30034, 40045}.
- d. RT2 swaps Binding-SID:100 to Segment list {30034, 40045}, and uses the 30034 to lookup the Next Label record of the Proxy Forwarding table again. The behavior found is to forward the packet to RT4.
- e. RT2 queries the Routing Table to RT4, using primary or backup path to RT4. The next hop is RT7.

f. RT2 forwards packets to RT7. RT7 queries the local routing table to forward the packet to RT4.

## 5. Security Considerations

The extensions to OSPF and IS-IS described in this document result in two types of behaviors in data plane when a node in a network fails. One is that for a node, which is a upstream (except for the direct upstream) node of the failed node along a SR-TE path, it continues to send the traffic to the failed node along the SR-TE path for an extended period of time. The other is that for a node, which is the direct upstream node of the failed node, it fast re-routes the traffic around the failed node to the direct downstream node of the failed node along the SR-TE path. These behaviors are internal to a network and should not cause extra security issues.

## 6. Acknowledgements

The authors would like to thank Peter Psenak, Acee Lindem, Les Ginsberg, Bruno Decraene and Jeff Tantsura for their comments to this work.

## 7. References

### 7.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC7356] Ginsberg, L., Previdi, S., and Y. Yang, "IS-IS Flooding Scope Link State PDUs (LSPs)", RFC 7356, DOI 10.17487/RFC7356, September 2014, <<https://www.rfc-editor.org/info/rfc7356>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8402] Filsfils, C., Ed., Previdi, S., Ed., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", RFC 8402, DOI 10.17487/RFC8402, July 2018, <<https://www.rfc-editor.org/info/rfc8402>>.

[RFC8667] Previdi, S., Ed., Ginsberg, L., Ed., Filsfils, C., Bashandy, A., Gredler, H., and B. Decraene, "IS-IS Extensions for Segment Routing", RFC 8667, DOI 10.17487/RFC8667, December 2019, <<https://www.rfc-editor.org/info/rfc8667>>.

## 7.2. Informative References

[I-D.ietf-rtgwg-segment-routing-ti-lfa]  
Litkowski, S., Bashandy, A., Filsfils, C., Francois, P., Decraene, B., and D. Voyer, "Topology Independent Fast Reroute using Segment Routing", Work in Progress, Internet-Draft, draft-ietf-rtgwg-segment-routing-ti-lfa-08, 21 January 2022, <<https://www.ietf.org/archive/id/draft-ietf-rtgwg-segment-routing-ti-lfa-08.txt>>.

[I-D.ietf-spring-segment-protection-sr-te-paths]  
Hegde, S., Bowers, C., Litkowski, S., Xu, X., and F. Xu, "Segment Protection for SR-TE Paths", Work in Progress, Internet-Draft, draft-ietf-spring-segment-protection-sr-te-paths-03, 7 March 2022, <<https://www.ietf.org/archive/id/draft-ietf-spring-segment-protection-sr-te-paths-03.txt>>.

[I-D.ietf-spring-segment-routing-policy]  
Filsfils, C., Talaulikar, K., Voyer, D., Bogdanov, A., and P. Mattes, "Segment Routing Policy Architecture", Work in Progress, Internet-Draft, draft-ietf-spring-segment-routing-policy-22, 22 March 2022, <<https://www.ietf.org/archive/id/draft-ietf-spring-segment-routing-policy-22.txt>>.

## Appendix A. Proxy Forwarding for Adjacency and Node Segment

This Section shows through example how a proxy node forward traffic to the destination node when a node fails and the next segment of label stack is an adjacency-SID or node-SID.

### A.1. Next Segment is an Adjacency Segment

As shown in Figure 1, Label Stack 3 {10012, 20023, 30034, 40045} uses only adjacency-SIDs and represents the SR-TE strict explicit path RT1->RT2->RT3->RT4->RT5. When RT3 fails, node RT2 acts as a PLR, and uses next adjacency-SID (30034) of the label stack to lookup the proxy forwarding table built by RT2 locally for RT3. The path returned is the label forwarding path to RT3's next hop node RT4, which bypasses RT3. The specific steps are as follows:

- a. RT1 pops top adjacency-SID 10012, and forwards the packet to RT2;
- b. RT2 uses the label 20023 to identify the next hop node RT3, which has failed. RT2 pops label 20023 and queries the Proxy Forwarding Table corresponding to RT3 with label 30034. The query result is 2004. RT2 uses 2004 as the incoming label to query the label forwarding table. The next hop is RT7, and the incoming label is changed to 7004.
- c. So the packet leaves RT2 out the interface to RT7 with label stack {7004, 40045}. RT7 forwards it to RT4, where the original path is rejoined.
- d. RT2 forwards packets to RT7. RT7 queries the local routing table to forward the packet to RT4.

#### A.2. Next Segment is a Node Segment

As shown in Figure 1, Label Stack 2 {1003, 3004, 4005} uses only node-SIDs and represents the ECMP-aware path RT1->RT3->RT4->RT5, where 1003 is the node SID of RT3.

When the node RT3 fails, the proxy forwarding TLV advertised by the RT2 is preferred to direct the traffic of the RT1 to the PLR node RT2. Node RT2 acts as a PLR node and queries the proxy forwarding table locally built for RT3. The path returned is the label forwarding path to RT3's next hop node RT4, which bypasses RT3. The specific steps are as follows:

- a. RT1 swaps label 1003 to out-label 2003 to RT3.
- b. RT2 receives the label forwarding packet whose top label of label stack is 2003, and searches for the local Routing Table, the behavior found is to lookup Proxy Forwarding table due to RT3 failure, RT2 pops label 2003.
- c. RT2 uses 3004 as the in-label to lookup Proxy Forwarding table, The value of Map Label calculated based on SRGBDiffValue is 2004. and the query result is forwarding the packet to RT4.
- d. Then RT2 queries the Routing Table to RT4, using the primary or backup path to RT4. The next hop is RT7.
- e. RT2 forwards the packet to RT7. RT7 queries the local routing table to forward the packet to RT4.
- f. After RT1 convergences, node SID 1003 is preferred to the proxy SID implied/advertised by RT2.

## Authors' Addresses

Zhibo Hu  
Huawei Technologies  
Huawei Bld., No.156 Beiqing Rd.  
Beijing  
100095  
China  
Email: huzhibo@huawei.com

Huaimo Chen  
Futurewei  
Boston, MA,  
United States of America  
Email: Huaimo.chen@futurewei.com

Junda Yao  
Huawei Technologies  
Huawei Bld., No.156 Beiqing Rd.  
Beijing  
100095  
China  
Email: yaojunda@huawei.com

Chris Bowers  
Juniper Networks  
1194 N. Mathilda Ave.  
Sunnyvale, CA, 94089  
United States of America  
Email: cbowers@juniper.net

Yongqing  
China Telecom  
109, West Zhongshan Road, Tianhe District  
Guangzhou  
510000  
China  
Email: zhuyq8@chinatelecom.cn

Yisong  
China Mobile  
510000  
China



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

RTGWG  
Internet-Draft  
Intended status: Standards Track  
Expires: May 6, 2020

F. Zheng  
B. Wu, Ed.  
Huawei  
R. Wilton, Ed.  
Cisco Systems  
X. Ding  
November 3, 2019

YANG Data Model for ARP  
draft-ietf-rtgwg-arp-yang-model-03

Abstract

This document defines a YANG data model for the management of the Address Resolution Protocol (ARP). It extends the basic ARP functionality contained in the ietf-ip YANG data model, defined in RFC 8344, to provide management of optional ARP features and statistics.

The YANG data model in this document conforms to the Network Management Datastore Architecture defined in RFC 8342.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

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."

This Internet-Draft will expire on May 6, 2020.

Copyright Notice

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 (<https://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.

## Table of Contents

|  |    |
|--|----|
| 1. Introduction . . . . .                                    | 2  |
| 1.1. Terminology . . . . .                                   | 3  |
| 1.2. Tree Diagrams . . . . .                                 | 4  |
| 2. Problem Statement . . . . .                               | 4  |
| 3. Design of the Data Model . . . . .                        | 4  |
| 3.1. ARP Dynamic Learning . . . . .                          | 4  |
| 3.2. Proxy ARP . . . . .                                     | 5  |
| 3.3. Gratuitous ARP . . . . .                                | 5  |
| 3.4. ARP Data Model . . . . .                                | 5  |
| 4. ARP YANG Module . . . . .                                 | 6  |
| 5. Data Model Examples . . . . .                             | 11 |
| 5.1. Configured static ARP Entry . . . . .                   | 11 |
| 5.2. Configuration of proxy ARP and gratuitous ARP . . . . . | 12 |
| 6. IANA Considerations . . . . .                             | 13 |
| 7. Security Considerations . . . . .                         | 13 |
| 8. Acknowledgments . . . . .                                 | 14 |
| 9. References . . . . .                                      | 14 |
| 9.1. Normative References . . . . .                          | 14 |
| 9.2. Informative References . . . . .                        | 16 |
| Authors' Addresses . . . . .                                 | 17 |

## 1. Introduction

Basic ARP functionality is supported by the ietf-ip YANG data model, defined in [RFC8344]. This document defines a YANG [RFC7950] data model that extends the basic ARP YANG support to also cover optional ARP features, and ARP related statistics to aid network monitoring and troubleshooting.

This model defines YANG configuration and operational state data nodes both for ARP related functionality formally specified in other RFCs (such as [RFC8344] and [RFC1027]), but also for common ARP behaviour that is often supported on network devices.

Where necessary, the expected behaviour of the YANG data nodes is defined in the YANG model, and this document.

The YANG modules in this document conform to the Network Management Datastore Architecture (NMDA) [RFC8342].

Editorial Note: (To be removed by RFC Editor)

This draft contains several placeholder values that need to be replaced with finalized values at the time of publication. Please apply the following replacements:

- o "XXXX" --> the assigned RFC value for this draft both in this draft and in the YANG models under the revision statement.
- o The "revision" date in model, in the format XXXX-XX-XX, needs to be updated with the date the draft gets approved. The date also needs to get reflected on the line with <CODE BEGINS>.

### 1.1. Terminology

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.

The following terms are defined in [RFC8342] and are not redefined here:

- o client
- o server
- o configuration data
- o system state
- o state data
- o intended configuration
- o running configuration datastore
- o operational state datastore

The following terms are defined in [RFC7950] and are not redefined here:

- o augment
- o data model
- o data node

The terminology for describing YANG data models is found in [RFC7950].

## 1.2. Tree Diagrams

Tree diagrams used in this document follow the notation defined in [RFC8340]

## 2. Problem Statement

Neither ARP [RFC0826], nor Proxy-ARP [RFC1027], define standard network management configuration models. Instead, network equipment vendors have implemented their own bespoke configuration interfaces and models.

Network operators benefit from having common network management models defined that can be implemented by multiple network equipment manufacturers. This simplifies the operation and management of network devices.

Some, but not all, required ARP functionality has been defined in ietf-ip.yang ([RFC8344]). Providing a standard YANG model that models these optional ARP features, that are fairly widely implemented by network equipment manufacturers, and used by network operators, is beneficial to the general goal of interoperability in the networking industry.

## 3. Design of the Data Model

This data model intends to describe the processing that a protocol finds the hardware address, also known as Media Access Control (MAC) address, of a host from its known IP address. These tasks include, but are not limited to, configuring dynamic ARP learning, proxy ARP, gratuitous ARP. There are two kind of ARP configurations: global ARP configuration, which is across all interfaces on the device, and per interface ARP configuration.

### 3.1. ARP Dynamic Learning

As defined in [RFC0826], ARP caching is the method of storing network addresses and the associated data-link addresses in memory for a period of time as the addresses are learned. This minimizes the use of valuable network resources to broadcast for the same address each time a datagram is sent.

There are static ARP cache entries and dynamic ARP cache entries. Static entries, are manually configured and kept in the cache table on a permanent basis which are defined in the ipv4 neighbor list for

each interface in [RFC8344]. Dynamic entries are added by vendor software, kept for a period of time, and then removed. We can specify how long an entry remains in the ARP cache. If we specify a timeout of 0 seconds, entries are never cleared from the ARP cache.

### 3.2. Proxy ARP

Proxy ARP, defined in [RFC1027], allows a router to respond to ARP requests on behalf of another machine that is not on the same local subnet, offering its own Ethernet media access control (MAC) address. By replying in such a way, the router then takes responsibility for routing packets to the intended destination.

In the case of certain data center network virtualization, as specified in [RFC8014], the proxy ARP can be extended to intercept all ARP requests, including source and target IP addresses in different subnets, and those ARP requests in the same subnet to suppress ARP handling.

### 3.3. Gratuitous ARP

Gratuitous ARP enables a device to send an ARP Request packet using its own IP address as the destination address. Gratuitous ARP provides the following functions:

- o Checks duplicate IP addresses: [RFC5227] uses gratuitous ARP to help detect IP conflicts. When a device receives an ARP request containing a source IP that matches its own, then it knows there is an IP conflict.
- o Advertises a new MAC address: Also in [RFC5227], if the MAC address of a host changes because its network adapter is replaced, the host sends a gratuitous ARP packet to notify all hosts of the change before the ARP entry is aged out.
- o Notifies an active/standby switchover in a [RFC5798] VRRP backup group: After an active/standby switchover, the master router sends a gratuitous ARP packet in the VRRP backup group to notify the switchover.

### 3.4. ARP Data Model

This document defines the YANG module "ietf-arp", which has the following structure:

```

module: ietf-arp
  +--rw arp
    +--rw dynamic-learning?    boolean

  augment /if:interfaces/if:interface/ip:ipv4:
    +--rw arp
      +--rw expiry-time?      uint32
      +--rw dynamic-learning? boolean
      +--rw proxy-arp
        | +--rw mode?          enumeration
      +--rw gratuitous-arp
        | +--rw enable?        boolean
        | +--rw interval?      uint32
      +--ro statistics
        +--ro in-requests-pkts? yang:counter32
        +--ro in-replies-pkts?  yang:counter32
        +--ro in-gratuitous-pkts? yang:counter32
        +--ro out-requests-pkts? yang:counter32
        +--ro out-replies-pkts?  yang:counter32
        +--ro out-gratuitous-pkts? yang:counter32
  augment /if:interfaces/if:interface/ip:ipv4/ip:neighbor:
    +--ro remaining-expiry-time? uint32

```

#### 4. ARP YANG Module

This section presents the ARP YANG module defined in this document.

This module imports definitions from Common YANG Data Types [RFC6991], A YANG Data Model for Interface Management [RFC8343], and A YANG Data Model for IP Management [RFC8344].

<CODE BEGINS> file "ietf-arp@2019-11-04.yang"

```

module ietf-arp {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-arp";
  prefix arp;

  import ietf-yang-types {
    prefix yang;
    reference "RFC 6991: Common YANG Data Types";
  }
  import ietf-interfaces {
    prefix if;
    reference "RFC 8343: A Yang Data Model for Interface Management";
  }
  import ietf-ip {
    prefix ip;
  }

```

```
    reference "RFC 8344: A Yang Data Model for IP Management";
  }

  organization
    "IETF Routing Area Working Group (rtgwg)";
  contact
    "WG Web: <http://tools.ietf.org/wg/rtgwg/>
    WG List: <mailto: rtgwg@ietf.org>
    Author: Feng Zheng
            hobby.zheng@huawei.com
    Editor: Bo Wu
            lana.wubo@huawei.com
    Editor: Robert Wilton
            rwilton@cisco.com
    Author: Xiaojian Ding
            wjswsl@163.com";
  description
    "Address Resolution Protocol (ARP) management, which includes
    static ARP configuration, dynamic ARP learning, ARP entry query,
    and packet statistics collection.

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

    Redistribution and use in source and binary forms, with or
    without modification, is permitted pursuant to, and subject
    to the license terms contained in, the Simplified BSD License
    set forth in Section 4.c of the IETF Trust's Legal Provisions
    Relating to IETF Documents
    (http://trustee.ietf.org/license-info).

    This version of this YANG module is part of RFC XXXX; see the
    RFC itself for full legal notices.";

  revision 2019-11-04 {
    description
      "Init revision";
    reference "RFC XXXX: A Yang Data Model for ARP";
  }

  container arp {
    description
      "Address Resolution Protocol (ARP)";
    leaf dynamic-learning {
      type boolean;
      default "true";
      description
        "Controls the default ARP learning behavior on all
```



```
        interfaces on the device, unless explicit overridden by
        the per-interface dynamic-learning leaf:
            true - dynamic learning is enabled on all interfaces by
                  default,
            false - dynamic learning is disabled on all interfaces by
                  default";
        reference "RFC826: An Ethernet Address Resolution Protocol";
    }
}
augment "/if:interfaces/if:interface/ip:ipv4" {
    description
        "Augment interfaces with ARP configuration and state.";
    container arp {
        description
            "Address Resolution Protocol (ARP) related configuration
            and state";
        leaf expiry-time {
            type uint32 {
                range "30..86400";
            }
            units "seconds";
            description
                "Aging time of a received dynamic ARP entry before it is
                removed from the cache.";
        }
        leaf dynamic-learning {
            type boolean;
            description
                "Controls whether dynamic ARP learning is enabled on the
                interface. If not configured, it defaults to the behavior
                specified in the per-device /arp/dynamic-learning leaf.

                true - dynamic learning is enabled
                false - dynamic learning is disabled";
        }
    }
    container proxy-arp {
        description
            "Configuration parameters for proxy ARP";
        leaf mode {
            type enumeration {
                enum disabled {
                    description
                        "The system only responds to ARP requests that
                        specify a target address configured on the local
                        interface.";
                }
                enum remote-only {
                    description

```

```
        "The system only responds to ARP requests when the
        sender and target IP addresses are in different
        subnets.";
    }
    enum all {
        description
            "The system responds to ARP requests where the sender
            and target IP addresses are in different subnets, as
            well as those where they are in the same subnet.";
    }
}
default "disabled";
description
    "When set to a value other than 'disable', the local
    system should respond to ARP requests that are for
    target addresses other than those that are configured on
    the local subinterface using its own MAC address as the
    target hardware address. If the 'remote-only' value is
    specified, replies are only sent when the target address
    falls outside the locally configured subnets on the
    interface, whereas with the 'all' value, all requests,
    regardless of their target address are replied to.";
reference
    "RFC1027: Using ARP to Implement Transparent Subnet
    Gateways";
}
}
container gratuitous-arp {
    description "Configure gratuitous ARP.";
    reference "RFC5227: IPv4 Address Conflict Detection";
    leaf enable {
        type boolean;
        description
            "Enable or disable sending gratuitous ARP packet on the
            interface.

            The default behaviour is device specific, and a
            deviation could used to to specify a device specific
            default.";
    }
    leaf interval {
        type uint32 {
            range "1..86400";
        }
        units "seconds";
        description
            "The interval, in seconds, between sending gratuitous ARP
            packet on the interface.
```

```
        The default behaviour is device specific, and a
        deviation could used to to specify a device specific
        default.";
    }
}
container statistics {
    config false;
    description
        "ARP per-interface packet statistics

        For all ARP interface counters, discontinuities in the
        value can occur at re-initialization of the management
        system and at other times as indicated by the value of
        '../../statistics/discontinuity-time' in the
        ietf-interfaces YANG module.";

    leaf in-requests-pkts {
        type yang:counter32;
        description
            "The number of ARP request packets received on this
            interface.";
    }

    leaf in-replies-pkts {
        type yang:counter32;
        description
            "The number of ARP reply packets received on this
            interface.";
    }

    leaf in-gratuitous-pkts {
        type yang:counter32;
        description
            "The number of gratuitous ARP packets received on this
            interface.";
    }

    leaf out-requests-pkts {
        type yang:counter32;
        description
            "The number of ARP request packets sent on this
            interface.";
    }

    leaf out-replies-pkts {
        type yang:counter32;
        description
            "The number of ARP reply packets sent on this
```

```
        interface.";
    }

    leaf out-gratuitous-pkts {
        type yang:counter32;
        description
            "The number of gratuitous ARP packets sent on this
            interface.";
    }
}

augment "/if:interfaces/if:interface/ip:ipv4/ip:neighbor" {
    description
        "Augment IPv4 neighbor list with ARP expiry time.";
    leaf remaining-expiry-time {
        type uint32;
        units "seconds";
        config false;
        description
            "The number of seconds until the dynamic ARP entry expires
            and is removed from the ARP cache.";
    }
}
```

## 5. Data Model Examples

This section presents two simple ARP configuration examples:

### 5.1. Configured static ARP Entry

This example illustrates the configuration for a static ARP entry for peer 192.0.2.1 with MAC address 00:00:5E:00:53:AB using the model defined in [RFC8344].

```
<?xml version="1.0" encoding="utf-8"?>
<interfaces
  xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces"
  xmlns:ianaift="urn:ietf:params:xml:ns:yang:iana-if-type">
  <interface>
    <name>eth0</name>
    <type>ianaift:ethernetCsmacd</type>
    <!-- other parameters from ietf-interfaces omitted -->

    <ipv4 xmlns="urn:ietf:params:xml:ns:yang:ietf-ip">
      <!-- ipv4 address configuration parameters omitted -->
      <neighbor>
        <ip>192.0.2.1</ip>
        <link-layer-address>00:00:5E:00:53:AB</link-layer-address>
      </neighbor>
    </ipv4>
  </interface>
</interfaces>
```

## 5.2. Configuration of proxy ARP and gratuitous ARP

This example illustrates the configuration of ARP entry expiry time, proxy ARP in 'remote-only' mode, and enabling gratuitous ARP with an interval of 10 minutes.

```
<?xml version="1.0" encoding="utf-8"?>
<interfaces
  xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces"
  xmlns:ianaift="urn:ietf:params:xml:ns:yang:iana-if-type">
  <interface>
    <name>eth0</name>
    <type>ianaift:ethernetCsmacd</type>
    <!-- other parameters from ietf-interfaces omitted -->

    <ipv4 xmlns="urn:ietf:params:xml:ns:yang:ietf-ip">
      <!-- ipv4 address configuration parameters omitted -->
      <arp xmlns="urn:ietf:params:xml:ns:yang:ietf-arp">
        <expiry-time>1200</expiry-time>
        <proxy-arp>
          <mode>remote-only</mode>
        </proxy-arp>
        <gratuitous-arp>
          <enable>true</enable>
          <interval>600</interval>
        </gratuitous-arp>
      </arp>
    </ipv4>
  </interface>
</interfaces>
```

## 6. IANA Considerations

This document registers a URI in the IETF XML registry [RFC3688]. Following the format in [RFC3688], the following registration is requested to be made:

URI: urn:ietf:params:xml:ns:yang:ietf-arp  
Registrant Contact: The RTGWG WG of the IETF.  
XML: N/A, the requested URI is an XML namespace.

This document registers a YANG module in the YANG Module Names registry [RFC6020].

Name: ietf-arp  
Namespace: urn:ietf:params:xml:ns:yang:ietf-arp  
Prefix: arp  
Reference: RFC XXXX

## 7. Security Considerations

The YANG module specified in this document defines a schema for data that is designed to be accessed via network management protocols such as NETCONF [RFC6241] or RESTCONF [RFC8040] . The lowest NETCONF

layer is the secure transport layer, and the mandatory-to-implement secure transport is Secure Shell (SSH) [RFC6242]. The lowest RESTCONF layer is HTTPS, and the mandatory-to-implement secure transport is TLS [RFC8446].

The NETCONF access control model [RFC8341] provides the means to restrict access for particular NETCONF or RESTCONF users to a preconfigured subset of all available NETCONF or RESTCONF protocol operations and content..

There are a number of data nodes defined in this YANG module that are writable/creatable/deletable (i.e., config true, which is the default). These data nodes may be considered sensitive or vulnerable in some network environments. Write operations (e.g., edit-config) to these data nodes without proper protection can have a negative effect on network operations. These are the subtrees and data nodes and their sensitivity/vulnerability:

arp/dynamic-learning: This leaf is used to enable ARP dynamic learning on all interfaces. ARP dynamic learning could allow an attacker to inject spoofed traffic into the network, e.g. denial-of-service attack.

interface/ipv4/arp/proxy-arp: These leaves are used to enable proxy ARP on an interface. They could allow traffic to be mis-configured (denial-of-service attack).

interface/ipv4/arp/gratuitous-arp: These leaves are used to enable sending gratuitous ARP packet on an interface. This configuration could allow an attacker to inject spoofed traffic into the network, e.g. man-in-the-middle attack. The default value for this data node is device specific, and hence users of this model MUST understand whether or not gratuitous ARP is enabled and whether this could constitute a security risk.

## 8. Acknowledgments

The authors wish to thank Alex Campbell, Reshad Rahman, Qin Wu, Tom Petch, Jeffrey Haas, and others for their helpful comments.

## 9. References

### 9.1. Normative References

- [RFC0826] Plummer, D., "An Ethernet Address Resolution Protocol: Or Converting Network Protocol Addresses to 48.bit Ethernet Address for Transmission on Ethernet Hardware", STD 37, RFC 826, DOI 10.17487/RFC0826, November 1982, <<https://www.rfc-editor.org/info/rfc826>>.
- [RFC1027] Carl-Mitchell, S. and J. Quarterman, "Using ARP to implement transparent subnet gateways", RFC 1027, DOI 10.17487/RFC1027, October 1987, <<https://www.rfc-editor.org/info/rfc1027>>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <<https://www.rfc-editor.org/info/rfc3688>>.
- [RFC5227] Cheshire, S., "IPv4 Address Conflict Detection", RFC 5227, DOI 10.17487/RFC5227, July 2008, <<https://www.rfc-editor.org/info/rfc5227>>.
- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <<https://www.rfc-editor.org/info/rfc6020>>.
- [RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <<https://www.rfc-editor.org/info/rfc6242>>.
- [RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, DOI 10.17487/RFC6991, July 2013, <<https://www.rfc-editor.org/info/rfc6991>>.
- [RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <<https://www.rfc-editor.org/info/rfc7950>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.



- [RFC8342] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture (NMDA)", RFC 8342, DOI 10.17487/RFC8342, March 2018, <<https://www.rfc-editor.org/info/rfc8342>>.
- [RFC8343] Bjorklund, M., "A YANG Data Model for Interface Management", RFC 8343, DOI 10.17487/RFC8343, March 2018, <<https://www.rfc-editor.org/info/rfc8343>>.
- [RFC8344] Bjorklund, M., "A YANG Data Model for IP Management", RFC 8344, DOI 10.17487/RFC8344, March 2018, <<https://www.rfc-editor.org/info/rfc8344>>.
- [RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <<https://www.rfc-editor.org/info/rfc8446>>.

## 9.2. Informative References

- [RFC5798] Nadas, S., Ed., "Virtual Router Redundancy Protocol (VRRP) Version 3 for IPv4 and IPv6", RFC 5798, DOI 10.17487/RFC5798, March 2010, <<https://www.rfc-editor.org/info/rfc5798>>.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <<https://www.rfc-editor.org/info/rfc6241>>.
- [RFC8014] Black, D., Hudson, J., Kreeger, L., Lasserre, M., and T. Narten, "An Architecture for Data-Center Network Virtualization over Layer 3 (NVO3)", RFC 8014, DOI 10.17487/RFC8014, December 2016, <<https://www.rfc-editor.org/info/rfc8014>>.
- [RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <<https://www.rfc-editor.org/info/rfc8040>>.
- [RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, RFC 8340, DOI 10.17487/RFC8340, March 2018, <<https://www.rfc-editor.org/info/rfc8340>>.
- [RFC8341] Bierman, A. and M. Bjorklund, "Network Configuration Access Control Model", STD 91, RFC 8341, DOI 10.17487/RFC8341, March 2018, <<https://www.rfc-editor.org/info/rfc8341>>.

Authors' Addresses

Feng Zheng  
Huawei  
101 Software Avenue, Yuhua District  
Nanjing, Jiangsu 210012  
China

Email: [habby.zheng@huawei.com](mailto:habby.zheng@huawei.com)

Bo Wu (editor)  
Huawei

Email: [lane.wubo@huawei.com](mailto:lane.wubo@huawei.com)

Robert Wilton (editor)  
Cisco Systems

Email: [rwilton@cisco.com](mailto:rwilton@cisco.com)

Xiaojian Ding

Email: [wjswsl@163.com](mailto:wjswsl@163.com)

Network Working Group  
Internet Draft  
Intended status: Informational  
Expires: January 26, 2021

L. Dunbar  
Futurewei  
A. Malis  
Malis Consulting  
C. Jacquenet  
Orange  
July 26, 2020

Networks Connecting to Hybrid Cloud DCs: Gap Analysis  
draft-ietf-rtgwg-net2cloud-gap-analysis-07

Abstract

This document analyzes the IETF routing area technical gaps that may affect the dynamic connection to workloads and applications hosted in hybrid Cloud Data Centers from enterprise premises.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

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/ietf/lid-abstracts.txt>

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

This Internet-Draft will expire on January 26, 2009.

## Copyright Notice

Copyright (c) 2020 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.

## Table of Contents

|   |    |
|---|----|
| 1. Introduction.....  | 3  |
| 2. Conventions used in this document.....                           | 3  |
| 3. Gap Analysis for Accessing Cloud Resources.....                  | 4  |
| 3.1. Multiple PEs connecting to virtual CPEs in Cloud DCs.....      | 6  |
| 3.2. Access Control for workloads in the Cloud DCs.....             | 6  |
| 3.3. NAT Traversal.....   | 7  |
| 3.4. BGP between PEs and remote CPEs via Internet.....              | 7  |
| 3.5. Multicast traffic from/to the remote edges.....                | 8  |
| 4. Gap Analysis of Traffic over Multiple Underlay Networks.....     | 9  |
| 5. Aggregating VPN paths and Internet paths.....                    | 10 |
| 5.1. Control Plane for Cloud Access via Heterogeneous Networks..... | 11 |
| 5.2. Using BGP UPDATE Messages.....                                 | 12 |
| 5.2.1. Lack ways to differentiate traffic in Cloud DCs.....         | 12 |
| 5.2.2. Miss attributes in Tunnel-Encap.....                         | 12 |
| 5.3. SECURE-EVPN/BGP-EDGE-DISCOVERY.....                            | 12 |
| 5.4. SECURE-L3VPN.....  | 13 |
| 5.5. Preventing attacks from Internet-facing ports.....             | 14 |
| 6. Gap Summary.....   | 14 |
| 7. Manageability Considerations.....                                | 15 |
| 8. Security Considerations.....                                     | 16 |
| 9. IANA Considerations.....   | 16 |
| 10. References.....   | 16 |
| 10.1. Normative References.....                                     | 16 |
| 10.2. Informative References.....                                   | 16 |
| 11. Acknowledgments.....  | 17 |

## 1. Introduction

[Net2Cloud-Problem] describes the problems enterprises face today when interconnecting their branch offices with dynamic workloads hosted in third party data centers (a.k.a. Cloud DCs). In particular, this document analyzes the available routing protocols to identify whether there are any gaps that may impede such interconnection which may for example justify additional specification effort to define proper protocol extensions.

For the sake of readability, an edge, C-PE, or CPE are used interchangeably throughout this document. More precisely:

- . Edge: may include multiple devices (virtual or physical);
- . C-PE: provider-owned edge, e.g. for SECURE-EVPN's PE-based BGP/MPLS VPN, where PE is the edge node;
- . CPE: device located in enterprise premises.

## 2. Conventions used in this document

Cloud DC: Third party Data Centers that usually host applications and workload owned by different organizations or tenants.

Controller: Used interchangeably with Overlay controller to manage overlay path creation/deletion and monitor the path conditions between sites.

CPE-Based VPN: Virtual Private Network designed and deployed from CPEs. This is to differentiate from most commonly used PE-based VPNs a la RFC 4364.

OnPrem: On Premises data centers and branch offices

### 3. Gap Analysis for Accessing Cloud Resources

Because of the ephemeral property of the selected Cloud DCs for specific workloads/Apps, an enterprise or its network service provider may not have direct physical connections to the Cloud DCs that are optimal for hosting the enterprise's specific workloads/Apps. Under those circumstances, an overlay network design can be an option to interconnect the enterprise's on-premises data centers & branch offices to its desired Cloud DCs.

However, overlay paths established over the public Internet can have unpredictable performance, especially over long distances. Therefore, it is highly desirable to minimize the distance or the number of segments that traffic had to be forwarded over the public Internet.

The Metro Ethernet Forum's Cloud Service Architecture [MEF-Cloud] also describes a use case of network operators using Overlay paths over an LTE network or the public Internet for the last mile access where the VPN service providers cannot always provide the required physical infrastructure.

In some scenarios, some overlay edge nodes may not be directly attached to the PEs that participate to the delivery and the operation of the enterprise's VPN.

When using an overlay network to connect the enterprise's sites to the workloads hosted in Cloud DCs, the existing C-PEs at enterprise's sites have to be upgraded to connect to the said overlay network. If the workloads hosted in Cloud DCs need to be connected to many sites, the upgrade process can be very expensive.

[Net2Cloud-Problem] describes a hybrid network approach that extends the existing MPLS-based VPNs to the Cloud DC Workloads over the access paths that are not under the VPN provider's control. To make it work properly, a small number of the PEs of the BGP/MPLS VPN can be designated to connect to the remote workloads via secure IPsec tunnels. Those designated PEs are shown as fPE (floating PE or smart PE) in Figure 3. Once the secure IPsec tunnels are established, the workloads hosted in Cloud DCs can be reached by the enterprise's VPN without upgrading all of the enterprise's CPEs. The

only CPE that needs to connect to the overlay network would be a virtualized CPE instantiated within the cloud DC.

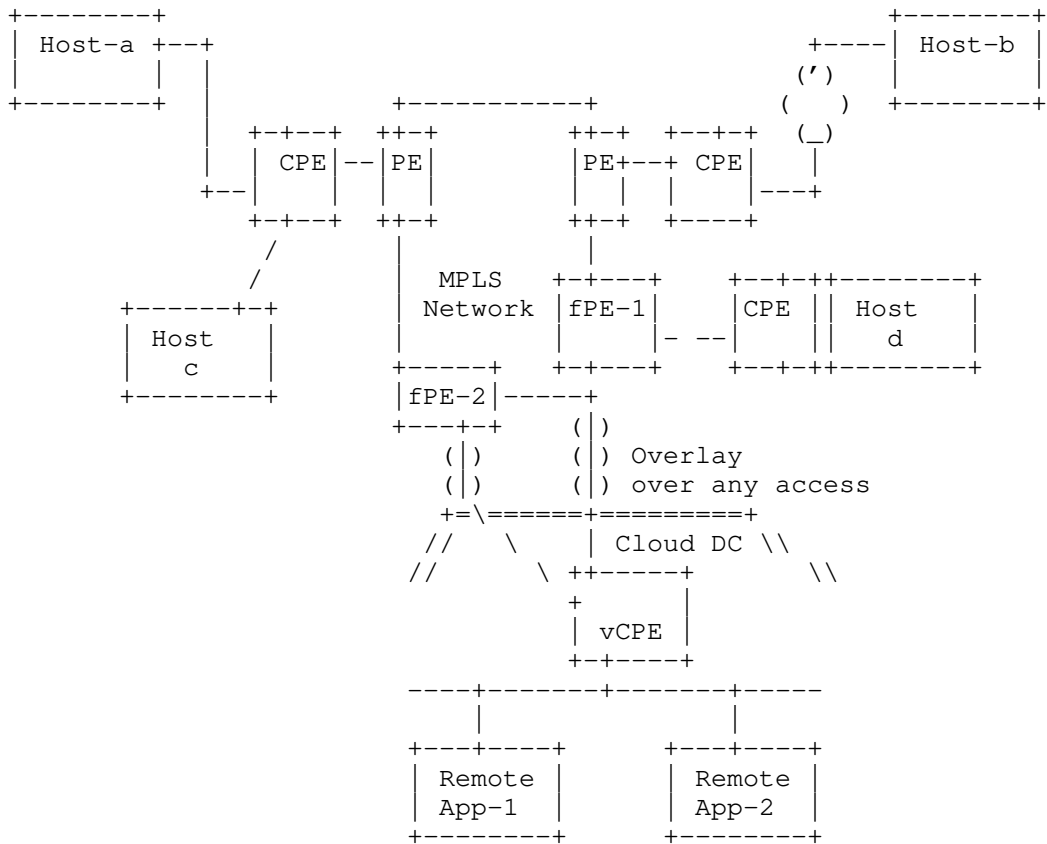


Figure 1: VPN Extension to Cloud DC

In Figure 1, the optimal Cloud DC to host the workloads (as a function of the proximity, capacity, pricing, or any other criteria chosen by the enterprises) does not have a direct connection to the PEs of the NGP/MPLS VPN that interconnects the enterprise's sites.

### 3.1. Multiple PEs connecting to virtual CPEs in Cloud DCs

To extend BGP/MPLS VPNs to virtual CPEs in Cloud DCs, it is necessary to establish secure tunnels (such as IPsec tunnels) between the PEs and the vCPEs.

Even though a set of PEs can be manually selected for a specific cloud data center, there are no standard protocols for those PEs to interact with the vCPEs instantiated in the third party cloud data centers over unsecure networks. The interaction includes exchanging performance, route information, etc..

When there is more than one PE available for use (as there should be for resiliency purposes or because of the need to support multiple cloud DCs geographically scattered), it is not straightforward to designate an egress PE to remote vCPEs based on applications. It might not be possible for PEs to recognize all applications because too much traffic traversing the PEs.

When there are multiple floating PEs that have established IPsec tunnels with a remote CPE, the remote CPE can forward outbound traffic to the optimal PE, which in turn forwards traffic to egress PEs to reach the final destinations. However, it is not straightforward for the ingress PE to select which egress PEs to send traffic. For example, in Figure 1:

- fPE-1 is the optimal PE for communication between App-1 <-> Host-a due to latency, pricing or other criteria.
- fPE-2 is the optimal PE for communication between App-1 <-> Host-b.

### 3.2. Access Control for workloads in the Cloud DCs

There is widespread diffusion of access policy for Cloud Resource, some of which is not easy for verification and validation. Because there are multiple parties involved in accessing Cloud Resources, policy enforcement points are not easily visible for policy refinement, monitoring, and testing.



The current state of the art for specifying access policies for Cloud Resources could be improved by having automated and reliable tools to map the user-friendly (natural language) rules into machine readable policies and to provide interfaces for enterprises to self-manage policy enforcement points for their own workloads.

### 3.3. NAT Traversal

Cloud DCs that only assign private IPv4 addresses to the instantiated workloads assume that traffic to/from the workload usually needs to traverse NATs.

There is no automatic way for an enterprise's network controller to be informed of the NAT properties for its workloads in Cloud DCs

One potential solution could be utilizing the messages sent during initialization of an IKE VPN when NAT Traversal option is enabled. There are some inherent problems while sending IPSec packets through NAT devices. One way to overcome these problems is to encapsulate IPSec packets in UDP. To do this effectively, there is a discovery phase in IKE (Phase1) that tries to determine if either of the IPSec gateways is behind a NAT device. If a NAT device is found, IPSec-over-UDP is proposed during IPSec (Phase 2) negotiation. If there is no NAT device detected, IPSec is used

Another potential solution could be allowing the virtual CPE in Cloud DCs to solicit a STUN (Session Traversal of UDP Through Network Address Translation, [RFC3489]) Server to get the information about the NAT property, the public IP addresses and port numbers so that such information can be communicated to the relevant peers.

### 3.4. BGP between PEs and remote CPEs via Internet

Even though an EBGp (external BGP) Multi-Hop design can be used to connect peers that are not directly connected to each other, there are still some issues about extending BGP from MPLS VPN PEs to remote CPEs in cloud DCs via any access path (e.g., Internet).

The path between the remote CPEs and VPN PEs that maintain VPN routes can traverse untrusted segments.

EBGP Multi-hop design requires configuration on both peers, either manually or via NETCONF from a controller. To use EBGP between a PE and remote CPEs, the PE has to be manually configured with the "next-hop" set to the IP address of the CPEs. When remote CPEs, especially remote virtualized CPEs are dynamically instantiated or removed, the configuration of Multi-Hop EBGP on the PE has to be changed accordingly.

Egress peering engineering (EPE) is not sufficient. Running BGP on virtualized CPEs in Cloud DCs requires GRE tunnels to be established first, which requires the remote CPEs to support address and key management capabilities. RFC 7024 (Virtual Hub & Spoke) and Hierarchical VPN do not support the required properties.

Also, there is a need for a mechanism to automatically trigger configuration changes on PEs when remote CPEs' are instantiated or moved (leading to an IP address change) or deleted.

EBGP Multi-hop design does not include a security mechanism by default. The PE and remote CPEs need secure communication channels when connecting via the public Internet.

Remote CPEs, if instantiated in Cloud DCs might have to traverse NATs to reach PEs. It is not clear how BGP can be used between devices located beyond the NAT and the devices located behind the NAT. It is not clear how to configure the Next Hop on the PEs to reach private IPv4 addresses.

### 3.5. Multicast traffic from/to the remote edges

Among the multiple floating PEs that are reachable from a remote CPE in a Cloud DC, multicast traffic sent by the remote CPE towards the MPLS VPN can be forwarded back to the remote CPE due to the PE receiving the multicast packets forwarding the multicast/broadcast frame to other PEs that in turn send to all attached CPEs. This process may cause traffic loops.

This problem can be solved by selecting one floating PE as the CPE's Designated Forwarder, similar to TRILL's Appointed Forwarders [RFC6325].

BGP/MPLS VPNs do not have features like TRILL's Appointed Forwarders.

#### 4. Gap Analysis of Traffic over Multiple Underlay Networks

Very often the Hybrid Cloud DCs are interconnected by multiple types of underlay networks, such as VPN, public Internet, wireless and wired infrastructures, etc. Sometimes the enterprises' VPN providers do not have direct access to the Cloud DCs that host some specific applications or workloads operated by the enterprise.

When reached by an untrusted network, all sensitive data to/from this virtual CPE have to be encrypted, usually by means of IPsec tunnels. When reached by a trusted direct connect paths, sensitive data can be forwarded without encryption for better performance.

If a virtual CPE in Cloud DC can be reached by both trusted and untrusted paths, better performance can be achieved to have a mixed encrypted and unencrypted traffic depending which paths the traffic is forwarded. However, there is no appropriate control plane protocol to achieve this automatically.

Some networks achieve the IPsec tunnel automation by using the modified NHRP protocol [RFC2332] to register network facing ports of the edge nodes with their Controller (or NHRP server), which then maps a private VPN address to a public IP address of the destination node/port. DSVPN [DSVPN] or DMVPN [DMVPN] are used to establish tunnels between WAN ports of SDWAN edge nodes.

NHRP was originally intended for ATM address resolution, and as a result, it misses many attributes that are necessary for dynamic virtual C-PE registration to the controller, such as:

- Interworking with the MPLS VPN control plane. An overlay edge can have some ports facing the MPLS VPN network over which packets can be forwarded without any encryption and some ports facing the

public Internet over which sensitive traffic needs to be encrypted.

- Scalability: NHRP/DSVPN/DMVPN work fine with small numbers of edge nodes. When a network has more than 100 nodes, these protocols do not scale well.
- NHRP does not have the IPsec attributes, which are needed for peers to build Security Associations over the public Internet.
- NHRP messages do not have any field to encode the C-PE supported encapsulation types, such as IPsec-GRE or IPsec-VxLAN.
- NHRP messages do not have any field to encode C-PE Location identifiers, such as Site Identifier, System ID, and/or Port ID.
- NHRP messages do not have any field to describe the gateway(s) to which the C-PE is attached. When a C-PE is instantiated in a Cloud DC, it is desirable for the C-PE's owner to be informed about how and where the C-PE is attached.
- NHRP messages do not have any field to describe C-PE's NAT properties if the C-PE is using private IPv4 addresses, such as the NAT type, Private address, Public address, Private port, Public port, etc.

## 5. Aggregating VPN paths and Internet paths

Most likely, enterprises (especially the largest ones) already have their C-PEs interconnected by VPNs, based upon VPN techniques like EVPN, L2VPN, or L3VPN. Their VPN providers might have direct paths/links to the Cloud DCs that host their workloads and applications.

When there is short term high traffic volume that can't justify increasing the VPNs capacity, enterprises can utilize public internet to reach their Cloud vCPEs. Then it is necessary for the vCPEs to communicate with the controller on how traffic is distributed among multiple heterogeneous underlay networks and to manage secure tunnels over untrusted networks.

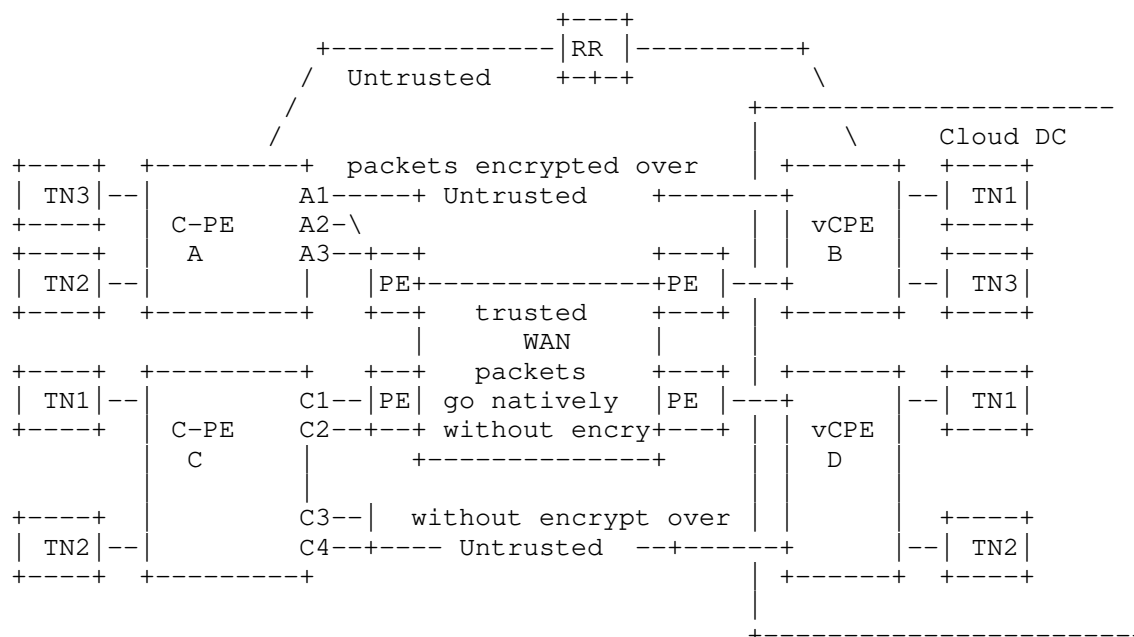


Figure 2: vCPEs reached by Hybrid Paths

### 5.1. Control Plane for Cloud Access via Heterogeneous Networks

The Control Plane for managing applications and workloads in cloud DCs reachable by heterogeneous networks need to include the following properties:

- vCPE in a cloud DCs needs to communicate with its controller of the properties of the directly connected underlay networks.
- Need Controller-facilitated IPsec SA attributes and NAT information distribution
  - o The controller facilitates and manages the peer authentication for all IPsec tunnels terminated at the vCPEs.
- Establishing and Managing the topology and reachability for services attached to the vCPEs in Cloud DCs.
  - o This is for the overlay layer's route distribution, so that a vCPE can populate its overlay routing table with

entries that identify the next hop for reaching a specific route/service attached to the vCPEs.

## 5.2. Using BGP UPDATE Messages

### 5.2.1. Lack ways to differentiate traffic in Cloud DCs

One enterprise can have different types of applications in one Cloud DC. Some can be production applications, some can be testing applications, and some can belong to one specific departments. The traffic to/from different applications might need to traverse different network paths or need to be differentiated by Control plane and data plane.

BGP already has built-in mechanisms, like Route Target, to differentiate different VPNs. But Route Target (RT) is for MPLS based VPNs, therefore RT is not appropriate to directly apply to virtual paths laid over mixed VPNs, IPsec or public underlay networks.

### 5.2.2. Miss attributes in Tunnel-Encap

[Tunnel-Encap] describes the BGP UPDATE Tunnel Path Attribute that advertises endpoints' tunnel encapsulation capabilities for the respective attached client routes encoded in the MP-NLRI Path Attribute. The receivers of the BGP UPDATE can use any of the supported encapsulations encoded in the Tunnel Path Attribute for the routes encoded in the MP-NLRI Path Attribute.

Here are some of the issues raised by using [Tunnel-Encap] to distribute the property of client routes be carried by mixed of hybrid networks:

- [Tunnel-Encap] doesn't have encoding methods to advertise that a route can be carried by mixed of IPsec tunnels and other already supported tunnels.
- The mechanism defined in [Tunnel-Encap] does not facilitate the exchange of IPsec SA-specific attributes.

## 5.3. SECURE-EVPN/BGP-EDGE-DISCOVERY

[SECURE-EVPN] describes a solution that utilize BGP as control plane for the Scenario #1 described in [BGP-SDWAN-Usage]. It relies upon a

BGP cluster design to facilitate the key and policy exchange among PE devices to create private pair-wise IPsec Security Associations. [Secure-EVPN] attaches all the IPsec SA information to the actual client routes.

[BGP-Edge-DISCOVERY] proposes BGP UPDATES from client routers only include the IPsec SA identifiers (ID) to reference the IPsec SA attributes being advertised by separate Underlay Property BGP UPDATE messages. If a client route can be encrypted by multiple IPsec SAs, then multiple IPsec SA IDs are included in the Tunnel-Encap Path attribute for the client route.

[BGP-Edge-DISCOVERY] proposes detailed IPsec SA attributes are advertised in a separate BGP UPDATE for the underlay networks.

[Secure-EVPN] and [BGP-Edge-Discovery] differs in the information included in the client routes. [Secure-EVPN] attaches all the IPsec SA information to the actual client routes, whereas the [BGP-Edge-Discovery] only includes the IPsec SA IDs for the client routes. The IPsec SA IDs used by [BGP-Edge-Discovery] is pointing to the SA-Information which are advertised separately, with all the SA-Information attached to routes which describe the SDWAN underlay, such as WAN Ports or Node address.

#### 5.4. SECURE-L3VPN

[SECURE-L3VPN] describes a method to enrich BGP/MPLS VPN [RFC4364] capabilities to allow some PEs to connect to other PEs via public networks. [SECURE-L3VPN] introduces the concept of Red Interface & Black Interface used by PEs, where the RED interfaces are used to forward traffic into the VPN, and the Black Interfaces are used between WAN ports through which only IPsec-formatted packets are forwarded to the Internet or to any other backbone network, thereby eliminating the need for MPLS transport in the backbone.

[SECURE-L3VPN] assumes PEs use MPLS over IPsec when sending traffic through the Black Interfaces.

[SECURE-L3VPN] is useful, but it misses the aspects of aggregating VPN and Internet underlays. In addition:

- The [SECURE-L3VPN] assumes that a CPE "registers" with the RR. However, it does not say how. It assumes that the remote CPEs are pre-configured with the IPsec SA manually. For overlay networks to connect Hybrid Cloud DCs, Zero Touch Provisioning is expected. Manual configuration is not an option.

- The [SECURE-L3VPN] assumes that C-PEs and RRs are connected via an IPsec tunnel. For management channel, TLS/DTLS is more economical than IPsec. The following assumption made by [SECURE-L3VPN] can be difficult to meet in the environment where zero touch provisioning is expected:

A CPE must also be provisioned with whatever additional information is needed in order to set up an IPsec SA with each of the red RRs

- IPsec requires periodic refreshment of the keys. The [SECURE-L3VPN] does not provide any information about how to synchronize the refreshment among multiple nodes.
- IPsec usually sends configuration parameters to two endpoints only and lets these endpoints negotiate the key. The [SECURE-L3VPN] assumes that the RR is responsible for creating/managing the key for all endpoints. When one endpoint is compromised, all other connections may be impacted.

#### 5.5. Preventing attacks from Internet-facing ports

When C-PEs have Internet-facing ports, additional security risks are raised.

To mitigate security risks, in addition to requiring Anti-DDoS features on C-PEs, it is necessary for C-PEs to support means to determine whether traffic sent by remote peers is legitimate to prevent spoofing attacks, in particular.

#### 6. Gap Summary

Here is the summary of the technical gaps discussed in this document:

- For Accessing Cloud Resources
  - a) Traffic Path Management: when a remote vCPE can be reached by multiple PEs of one provider VPN network, it is not



straightforward to designate which egress PE to the remote vCPE based on applications or performance.

- b) NAT Traversal: There is no automatic way for an enterprise's network controller to be informed of the NAT properties for its workloads in Cloud DCs.
- c) There is no loop prevention for the multicast traffic to/from remote vCPE in Cloud DCs.

Needs a feature like Appointed Forwarder specified by TRILL to prevent multicast data frames from looping around.

- d) BGP between PEs and remote CPEs via untrusted networks.

- Missing control plane to manage the propagation of the property of networks connected to the virtual nodes in Cloud DCs.

BGP UPDATE propagate client's routes information, but don't distinguish underlay networks.

- Issues of aggregating traffic over private paths and Internet paths

- a) Control plane messages for different overlay segmentations needs to be differentiated. User traffic belonging to different segmentations need to be differentiated.
- b) BGP Tunnel Encap doesn't have ways to indicate a route or prefix that can be carried by both IPsec tunnels and VPN tunnels
- c) Missing clear methods in preventing attacks from Internet-facing ports

## 7. Manageability Considerations

Zero touch provisioning of overlay networks to interconnect Hybrid Clouds is highly desired. It is necessary for a newly powered up edge node to establish a secure connection (by means of TLS, DTLS, etc.) with its controller.

## 8. Security Considerations

Cloud Services are built upon shared infrastructures, therefore not secure by nature.

Secure user identity management, authentication, and access control mechanisms are important. Developing appropriate security measurements can enhance the confidence needed by enterprises to fully take advantage of Cloud Services.

## 9. IANA Considerations

This document requires no IANA actions. RFC Editor: Please remove this section before publication.

## 10. References

### 10.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

### 10.2. Informative References

[RFC8192] S. Hares, et al, "Interface to Network Security Functions (I2NSF) Problem Statement and Use Cases", July 2017

[RFC5521] P. Mohapatra, E. Rosen, "The BGP Encapsulation Subsequent Address Family Identifier (SAFI) and the BGP Tunnel Encapsulation Attribute", April 2009.

[BGP-EDGE-DISCOVERY] L. Dunbar, et al, "BGP UPDATE for SDWAN Edge Discovery ", draft-dunbar-idr-sdwan-edge-discovery-00, Work-in-progress, July 2020.

[BGP-SDWAN-Usage] L. Dunbar, et al, "BGP Usage for SDWAN Overlay Networks ", draft-dunbar-bess-bgp-sdwan-usage-08, work-in-progress, July 2020.

- [Tunnel-Encap] K. Patel, et al, "The BGP Tunnel Encapsulation Attribute", draft-ietf-idr-tunnel-encaps-17, July 2020.
- [SECURE-EVPN] A. Sajassi, et al, draft-sajassi-bess-secure-evpn-01, work in progress, March 2019.
- [SECURE-L3VPN] E. Rosen, "Provide Secure Layer L3VPNs over Public Infrastructure", draft-rosen-bess-secure-l3vpn-00, work-in-progress, July 2018
- [DMVPN] Dynamic Multi-point VPN:  
<https://www.cisco.com/c/en/us/products/security/dynamic-multipoint-vpn-dmvpn/index.html>
- [DSVPN] Dynamic Smart VPN:  
<http://forum.huawei.com/enterprise/en/thread-390771-1-1.html>
- [ITU-T-X1036] ITU-T Recommendation X.1036, "Framework for creation, storage, distribution and enforcement of policies for network security", Nov 2007.
- [Net2Cloud-Problem] L. Dunbar and A. Malis, "Seamless Interconnect Underlay to Cloud Overlay Problem Statement", draft-dm-net2cloud-problem-statement-02, June 2018

## 11. Acknowledgments

Acknowledgements to John Drake for his review and contributions. Many thanks to John Scudder for stimulating the clarification discussion on the Tunnel-Encap draft so that our gap analysis can be more accurate.

This document was prepared using 2-Word-v2.0.template.dot.

Authors' Addresses

Linda Dunbar  
Futurewei  
Email: ldunbar@futurewei.com

Andrew G. Malis  
Malis Consulting  
Email: agmalis@gmail.com

Christian Jacquenet  
Orange  
Rennes, 35000  
France  
Email: Christian.jacquenet@orange.com



Network Working Group  
Internet Draft  
Intended status: Informational  
Expires: September 3, 2022

L. Dunbar  
Futurewei  
Andy Malis  
Malis Consulting  
C. Jacquenet  
Orange  
M. Toy  
Verizon  
March 7, 2022

Dynamic Networks to Hybrid Cloud DCs Problem Statement  
draft-ietf-rtgwg-net2cloud-problem-statement-12

Abstract

This document describes the problems that enterprises face today when interconnecting their branch offices with dynamic workloads in third party data centers (a.k.a. Cloud DCs). There can be many problems associated with network connecting to or among Clouds, many of which probably are out of the IETF scope. The objective of this document is to identify some of the problems that need additional work in IETF Routing area. Other problems are out of the scope of this document.

This document focuses on the network problems that many enterprises face when they have workloads & applications & data split among different data centers, especially for those enterprises with multiple sites that are already interconnected by VPNs (e.g., MPLS L2VPN/L3VPN).

Current operational problems are examined to determine whether there is a need to improve existing protocols or whether a new protocol is necessary to solve them.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

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/ietf/lid-abstracts.txt>

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

This Internet-Draft will expire on September 7, 2022.

#### Copyright Notice

Copyright (c) 2022 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.

#### Table of Contents

|   |   |
|---|---|
| 1. Introduction.....  | 3 |
| 1.1. Key Characteristics of Cloud Services:.....                | 3 |
| 1.2. Connecting to Cloud Services.....                          | 3 |
| 1.3. Reaching App instances in the optimal Cloud DC locations.. | 4 |
| 2. Definition of terms.....                                     | 5 |
| 3. High Level Issues of Connecting to Multi-Cloud.....          | 6 |
| 3.1. 5G Edge Clouds.....  | 6 |
| 3.2. Security Issues.....                                       | 6 |
| 3.3. Authorization and Identity Management.....                 | 7 |

|  |    |
|--|----|
| 3.4. API abstraction.....  | 7  |
| 3.5. DNS for Cloud Resources.....                                  | 8  |
| 3.6. NAT for Cloud Services.....                                   | 9  |
| 3.7. Cloud Discovery.....  | 10 |
| 4. Interconnecting Enterprise Sites with Cloud DCs.....            | 10 |
| 4.1. Sites to Cloud DC.....  | 10 |
| 4.2. Inter-Cloud Interconnection.....                              | 12 |
| 5. Edge Clouds.....  | 14 |
| 6. Problems with MPLS-based VPNs extending to Hybrid Cloud DCs.... | 14 |
| 7. Problem with using IPsec tunnels to Cloud DCs.....              | 15 |
| 7.1. Scaling Issues with IPsec Tunnels.....                        | 16 |
| 7.2. Poor performance over long distance.....                      | 16 |
| 8. End-to-End Security Concerns for Data Flows.....                | 16 |
| 9. Requirements for Dynamic Cloud Data Center VPNs.....            | 17 |
| 10. Security Considerations.....                                   | 17 |
| 11. IANA Considerations.....                                       | 18 |
| 12. References.....  | 18 |
| 12.1. Normative References.....                                    | 18 |
| 12.2. Informative References.....                                  | 18 |
| 13. Acknowledgments.....   | 18 |

## 1. Introduction

### 1.1. Key Characteristics of Cloud Services:

Key characteristics of Cloud Services are on-demand, scalable, highly available, and usage-based billing. Cloud Services, such as, compute, storage, network functions (most likely virtual), third party managed applications, etc. are usually hosted and managed by third parties Cloud Operators. Here are some examples of Cloud network functions: Virtual Firewall services, Virtual private network services, Virtual PBX services including voice and video conferencing systems, etc. Cloud Data Center (DC) is shared infrastructure that hosts the Cloud Services to many customers.

### 1.2. Connecting to Cloud Services

With the advent of widely available third-party cloud DCs and services in diverse geographic locations and the advancement of tools for monitoring and predicting application behaviors, it is very attractive for enterprises to instantiate applications and workloads in locations that are geographically closest to their end-users. Such proximity can improve end-to-end latency and overall user experience. Conversely, an enterprise can easily shutdown



applications and workloads whenever end-users are in motion (thereby modifying the networking connection of subsequently relocated applications and workloads). In addition, enterprises may wish to take advantage of more and more business applications offered by cloud operators.

The networks that interconnect hybrid cloud DCs must address the following requirements:

- to access all workloads in the desired cloud DCs:  
Many enterprises include cloud in their disaster recovery strategy, such as enforcing periodic backup policies within the cloud, or running backup applications in the Cloud.
- Global reachability from different geographical zones, thereby facilitating the proximity of applications as a function of the end users' location, to improve latency.
- Elasticity: prompt connection to newly instantiated applications at Cloud DCs when usages increase and prompt release of connection after applications at locations being removed when demands change.
- Scalable policy management: apply the appropriate policies to the newly instantiated application instances at any Cloud DC location.

### 1.3. Reaching App instances in the optimal Cloud DC locations

Many applications have multiple instances instantiated in different Cloud DCs. The current state of the art solutions is typically based on DNS assisted with load balancer by responding a FQDN (Fully Qualified Domain Name) inquiry with an IP address of the closest or lowest cost DC that can reach the instance. Here are some problems associated with DNS based solutions:

- Dependent on client behavior
  - Client can cache results indefinitely
  - Client may not receive service even though there are servers available (before cache timeout) in other Cloud DCs.

- No inherent leverage of proximity information present in the network (routing) layer, resulting in loss of performance
  - Client on the west coast can be mapped to a DC on the east coast
- Inflexible traffic control:
  - Local DNS resolver become the unit of traffic management. This requires DNS to receive periodical update of the network condition, which is difficult.

## 2. Definition of terms

**Cloud DC:** Third party Data Centers that usually host applications and workload owned by different organizations or tenants.

**Controller:** Used interchangeably with SD-WAN controller to manage SD-WAN overlay path creation/deletion and monitoring the path conditions between two or more sites.

**DSVPN:** Dynamic Smart Virtual Private Network. DSVPN is a secure network that exchanges data between sites without needing to pass traffic through an organization's headquarter virtual private network (VPN) server or router.

**Heterogeneous Cloud:** applications and workloads split among Cloud DCs owned or managed by different operators.

**Hybrid Clouds:** Hybrid Clouds refers to an enterprise using its own on-premises DCs in addition to Cloud services provided by one or more cloud operators. (e.g. AWS, Azure, Google, Salesforces, SAP, etc).

**VPC:** Virtual Private Cloud is a virtual network dedicated to one client account. It is logically isolated from other virtual networks in a Cloud DC. Each client can launch his/her desired resources, such as compute, storage, or network functions into his/her VPC. Most Cloud

operators' VPCs only support private addresses, some support IPv4 only, others support IPv4/IPv6 dual stack.

### 3. High Level Issues of Connecting to Multi-Cloud

There are many problems associated with connecting to hybrid Cloud Services, many of which are out of the IETF scope. This section is to identify some of the high-level problems that can be addressed by IETF, especially by Routing area. Other problems are out of the scope of this document. By no means has this section covered all problems for connecting to Hybrid Cloud Services, e.g. difficulty in managing cloud spending is not discussed here.

#### 3.1. 5G Edge Clouds

5G edge cloud data centers have routers connecting to the 5G Core functions, such as Radio Control Functions, Session Management Function (SMF), Access Mobility Functions (AMF), User Plane Functions (UPF), etc. Those functions need to be connected to the Radio Data Unit (R-DU) on the Cell Tower. The UPFs need to be connected to the 5G Local Data Networks' ingress routers which might co-located the cloud edge data centers.

In addition, the 5G edge cloud data centers may host edge computing servers for Ultra-low latency services that need to be near the UEs (User equipment). Those edge computing applications need to have very low latency to the UEs, and also connect to backend servers or databases in another location.

#### 3.2. Security Issues

Cloud Services is built upon shared infrastructure, therefore not secure by nature. Security has been a primary, and valid, concern from the start of cloud computing, e.g. not being able to see the exact location where the data are stored or trace of access. Headlines highlighting data breaches, compromised credentials, and broken authentication, hacked interfaces and APIs, account hijacking haven't helped alleviate concerns.

Many Cloud operators offer monitoring services for data stored in Clouds, such as AWS CloudTrail, Azure Monitor, and many third-party monitoring tools to improve visibility to data stored in Clouds. But

there is still underline security concerns on illegitimate data and workloads access.

Secure user identity management, authentication, and access control mechanisms are important. Developing appropriate security measurements can enhance the confidence needed by enterprises to fully take advantage of Cloud Services.

### 3.3. Authorization and Identity Management

One of the more prominent challenges for Cloud Services is Identity Management and Authorization. The Authorization not only includes user authorization, but also the authorization of API calls by applications from different Cloud DCs managed by different Cloud Operators. In addition, there are authorization for Workload Migration, Data Migration, and Workload Management.

There are many types of users in cloud environments, e.g. end users for accessing applications hosted in Cloud DCs, Cloud-resource users who are responsible for setting permissions for the resources based on roles, access lists, IP addresses, domains, etc.

There are many types of Cloud authorizations: including MAC (Mandatory Access Control) - where each app owns individual access permissions, DAC (Discretionary Access Control) - where each app requests permissions from an external permissions app, RBAC (Role-based Access Control) - where the authorization service owns roles with different privileges on the cloud service, and ABAC (Attribute-based Access Control) - where access is based on request attributes and policies.

IETF hasn't yet developed comprehensive specification for Identity management and data models for Cloud Authorizations.

### 3.4. API abstraction

Different Cloud Operators have different APIs to access their Cloud resources, security functions, the NAT, etc.

It is difficult to move applications built by one Cloud operator's APIs to another. However, it is highly desirable to have a single and consistent way to manage the networks and respective security policies for interconnecting applications hosted in different Cloud DCs.

The desired property would be having a single network fabric to which different Cloud DCs and enterprise's multiple sites can be attached or detached, with a common interface for setting desired policies.

The difficulty of connecting applications in different Clouds might be stemmed from the fact that they are direct competitors. Usually traffic flow out of Cloud DCs incur charges. Therefore, direct communications between applications in different Cloud DCs can be more expensive than intra Cloud communications.

It is desirable to have a common API shim layer or abstraction for different Cloud providers to make it easier to move applications from one Cloud DC to another.

### 3.5. DNS for Cloud Resources

DNS name resolution is essential for on-premises and cloud-based resources. For customers with hybrid workloads, which include on-premises and cloud-based resources, extra steps are necessary to configure DNS to work seamlessly across both environments.

Cloud operators have their own DNS to resolve resources within their Cloud DCs and to well-known public domains. Cloud's DNS can be configured to forward queries to customer managed authoritative DNS servers hosted on-premises, and to respond to DNS queries forwarded by on-premises DNS servers.

For enterprises utilizing Cloud services by different cloud operators, it is necessary to establish policies and rules on how/where to forward DNS queries to. When applications in one Cloud need to communication with applications hosted in another Cloud, there could be DNS queries from one Cloud DC being forwarded to the enterprise's on-premise DNS, which in turn be forwarded to the DNS service in another Cloud. Needless to say, configuration can be complex depending on the application communication patterns.

However, even with carefully managed policies and configurations, collisions can still occur. If you use an internal name like `.cloud` and then want your services to be available via or within some other cloud provider which also uses `.cloud`, then it can't work. Therefore, it is better to use the global domain name even when an organization does not make all its namespace globally resolvable. An organization's globally unique DNS can include subdomains that cannot be resolved at all outside certain restricted paths, zones that resolve differently based on the origin of the query, and zones that resolve the same globally for all queries from any source.

Globally unique names do not equate to globally resolvable names or even global names that resolve the same way from every perspective. Globally unique names do prevent any possibility of collision at the present or in the future and they make DNSSEC trust manageable. Consider using a registered and fully qualified domain name (FQDN) from global DNS as the root for enterprise and other internal namespaces.

### 3.6. NAT for Cloud Services

Cloud resources, such as VM instances, are usually assigned with private IP addresses. By configuration, some private subnets can have the NAT function to reach out to external network and some private subnets are internal to Cloud only.

Different Cloud operators support different levels of NAT functions. For example, AWS NAT Gateway does not currently support connections towards, or from VPC Endpoints, VPN, AWS Direct Connect, or VPC Peering. <https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/vpc-nat-gateway.html#nat-gateway-other-services>. AWS Direct Connect/VPN/VPC Peering does not currently support any NAT functionality.

Google's Cloud NAT allows Google Cloud virtual machine (VM) instances without external IP addresses and private Google Kubernetes Engine (GKE) clusters to connect to the Internet. Cloud NAT implements outbound NAT in conjunction with a default route to allow instances to reach the Internet. It does not implement inbound NAT. Hosts outside of VPC network can only respond to established connections initiated by instances inside the Google Cloud; they cannot initiate their own, new connections to Cloud instances via NAT.

For enterprises with applications running in different Cloud DCs, proper configuration of NAT has to be performed in Cloud DC and in their on-premises DC.

### 3.7. Cloud Discovery

One of the concerns of using Cloud services is not aware where the resource is located, especially Cloud operators can move application instances from one place to another. When applications in Cloud communicate with on-premise applications, it may not be clear where the Cloud applications are located or to which VPCs they belong.

It is highly desirable to have tools to discover cloud services in much the same way as you would discover your on-premises infrastructure. A significant difference is that cloud discovery uses the cloud vendor's API to extract data on your cloud services, rather than the direct access used in scanning your on-premises infrastructure.

Standard data models, APIs or tools can alleviate concerns of enterprise utilizing Cloud Resources, e.g. having a Cloud service scan that connects to the API of the cloud provider and collects information directly.

## 4. Interconnecting Enterprise Sites with Cloud DCs

Considering that many enterprises already have existing VPNs (e.g. MPLS based L2VPN or L3VPN) interconnecting branch offices & on-premises data centers, connecting to Cloud services will be mixed of different types of networks. When an enterprise's existing VPN service providers do not have direct connections to the corresponding cloud DCs that the enterprise prefers to use, the enterprise has to face additional infrastructure and operational costs to utilize the Cloud services.

### 4.1. Sites to Cloud DC

Most Cloud operators offer some type of network gateway through which an enterprise can reach their workloads hosted in the Cloud DCs. AWS (Amazon Web Services) offers the following options to reach workloads in AWS Cloud DCs:

- AWS Internet gateway allows communication between instances in AWS VPC and the internet.
- AWS Virtual gateway (vGW) where IPsec tunnels [RFC6071] are established between an enterprise's own gateway and AWS vGW, so that the communications between those gateways can be secured from the underlay (which might be the public Internet).
- AWS Direct Connect, which allows enterprises to purchase direct connect from network service providers to get a private leased line interconnecting the enterprises gateway(s) and the AWS Direct Connect routers. In addition, an AWS Transit Gateway can be used to interconnect multiple VPCs in different Availability Zones. AWS Transit Gateway acts as a hub that controls how traffic is forwarded among all the connected networks which act like spokes.

Microsoft's ExpressRoute allows extension of a private network to any of the Microsoft cloud services, including Azure and Office365. ExpressRoute is configured using Layer 3 routing. Customers can opt for redundancy by provisioning dual links from their location to two Microsoft Enterprise edge routers (MSEEs) located within a third-party ExpressRoute peering location. The BGP routing protocol is then setup over WAN links to provide redundancy to the cloud. This redundancy is maintained from the peering data center into Microsoft's cloud network.

Google's Cloud Dedicated Interconnect offers similar network connectivity options as AWS and Microsoft. One distinct difference, however, is that Google's service allows customers access to the entire global cloud network by default. It does this by connecting your on-premises network with the Google Cloud using BGP and Google Cloud Routers to provide optimal paths to the different regions of the global cloud infrastructure.

Figure below shows an example of some of a tenant's workloads are accessible via a virtual router connected by AWS Internet Gateway; some are accessible via AWS vGW, and others are accessible via AWS Direct Connect.

Different types of access require different level of security functions. Sometimes it is not visible to end customers which type of network access is used for a specific application instance. To get better visibility, separate virtual routers (e.g. vR1 & vR2) can be deployed to differentiate traffic to/from different cloud GWs. It



is important for some enterprises to be able to observe the specific behaviors when connected by different connections.

Customer Gateway can be customer owned router or ports physically connected to AWS Direct Connect GW.

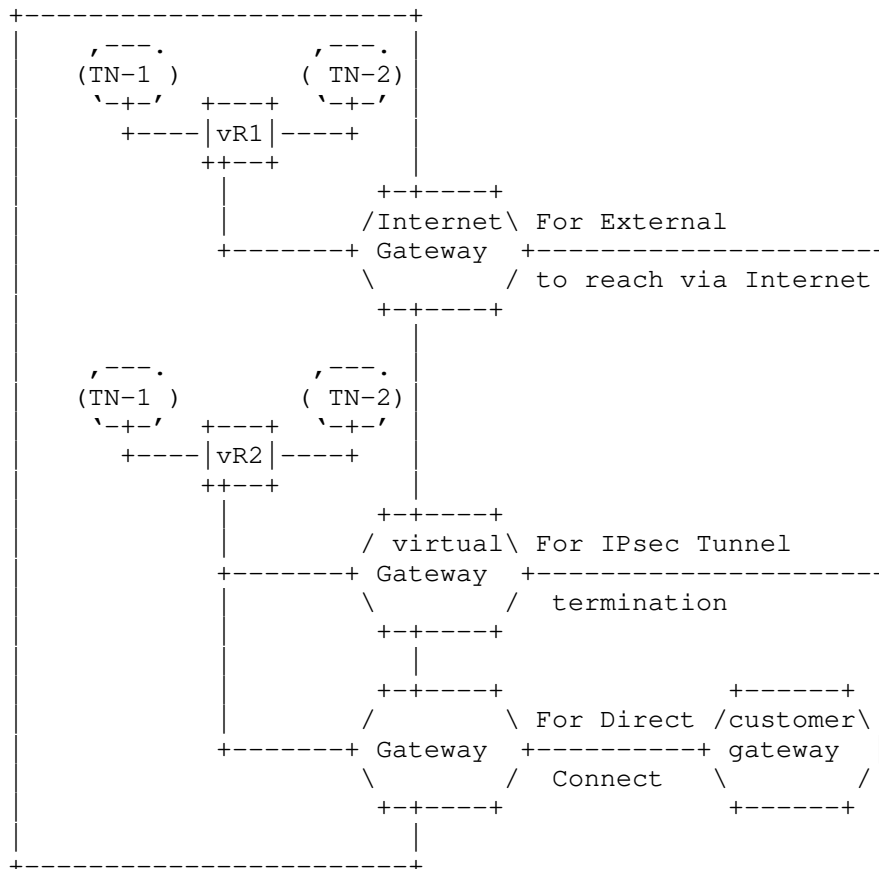


Figure 1: Examples of Multiple Cloud DC connections.

#### 4.2. Inter-Cloud Interconnection

The connectivity options to Cloud DCs described in the previous section are for reaching Cloud providers' DCs, but not between cloud DCs. When applications in AWS Cloud need to communicate with applications in Azure, today's practice requires a third-party gateway (physical or virtual) to interconnect the AWS's Layer 2 DirectConnect path with Azure's Layer 3 ExpressRoute.

Enterprises can also instantiate their own virtual routers in different Cloud DCs and administer IPsec tunnels among them, which by itself is not a trivial task. Or by leveraging open source VPN software such as strongSwan, you create an IPsec connection to the Azure gateway using a shared key. The StrongSwan instance within AWS not only can connect to Azure but can also be used to facilitate traffic to other nodes within the AWS VPC by configuring forwarding and using appropriate routing rules for the VPC.

Most Cloud operators, such as AWS VPC or Azure VNET, use non-globally routable CIDR from private IPv4 address ranges as specified by RFC1918. To establish IPsec tunnel between two Cloud DCs, it is necessary to exchange Public routable addresses for applications in different Cloud DCs.

In summary, here are some approaches, available now (which might change in the future), to interconnect workloads among different Cloud DCs:

- a) Utilize Cloud DC provided inter/intra-cloud connectivity services (e.g., AWS Transit Gateway) to connect workloads instantiated in multiple VPCs. Such services are provided with the cloud gateway to connect to external networks (e.g., AWS DirectConnect Gateway).
- b) Hairpin all traffic through the customer gateway, meaning all workloads are directly connected to the customer gateway, so that communications among workloads within one Cloud DC must traverse through the customer gateway.
- c) Establish direct tunnels among different VPCs (AWS' Virtual Private Clouds) and VNET (Azure's Virtual Networks) via client's own virtual routers instantiated within Cloud DCs. DMVPN (Dynamic Multipoint Virtual Private Network) or DSVPN (Dynamic Smart VPN) techniques can be used to establish direct Multi-point-to-Point or multi-point-to multi-point tunnels among those client's own virtual routers.

Approach a) usually does not work if Cloud DCs are owned and managed by different Cloud providers.

Approach b) creates additional transmission delay plus incurring cost when exiting Cloud DCs.

For the Approach c), DMVPN or DSVPN use NHRP (Next Hop Resolution Protocol) [RFC2735] so that spoke nodes can register their IP

addresses & WAN ports with the hub node. The IETF ION (Internetworking over NBMA (non-broadcast multiple access) WG standardized NHRP for connection-oriented NBMA network (such as ATM) network address resolution more than two decades ago.

There are many differences between virtual routers in Public Cloud DCs and the nodes in an NBMA network. NHRP cannot be used for registering virtual routers in Cloud DCs unless an extension of such protocols is developed for that purpose, e.g. taking NAT or dynamic addresses into consideration. Therefore, DMVPN and/or DSVPN cannot be used directly for connecting workloads in hybrid Cloud DCs.

## 5. Edge Clouds

## 6. Problems with MPLS-based VPNs extending to Hybrid Cloud DCs

Traditional MPLS-based VPNs have been widely deployed as an effective way to support businesses and organizations that require network performance and reliability. MPLS shifted the burden of managing a VPN service from enterprises to service providers. The CPEs attached to MPLS VPNs are also simpler and less expensive, because they do not need to manage routes to remote sites; they simply pass all outbound traffic to the MPLS VPN PEs to which the CPEs are attached (albeit multi-homing scenarios require more processing logic on CPEs). MPLS has addressed the problems of scale, availability, and fast recovery from network faults, and incorporated traffic-engineering capabilities.

However, traditional MPLS-based VPN solutions are sub-optimized for connecting end-users to dynamic workloads/applications in cloud DCs because:

- The Provider Edge (PE) nodes of the enterprise's VPNs might not have direct connections to third party cloud DCs that are used for hosting workloads with the goal of providing an easy access to enterprises' end-users.
- It takes some time to deploy provider edge (PE) routers at new locations. When enterprise's workloads are changed from one cloud DC to another (i.e., removed from one DC and re-instantiated to another location when demand changes), the

enterprise branch offices need to be connected to the new cloud DC, but the network service provider might not have PEs located at the new location.

One of the main drivers for moving workloads into the cloud is the widely available cloud DCs at geographically diverse locations, where apps can be instantiated so that they can be as close to their end-users as possible. When the user base changes, the applications may be migrated to a new cloud DC location closest to the new user base.

- Most of the cloud DCs do not expose their internal networks. An enterprise with a hybrid cloud deployment can use an MPLS-VPN to connect to a Cloud provider at multiple locations. The connection locations often correspond to gateways of different Cloud DC locations from the Cloud provider. The different Cloud DCs are interconnected by the Cloud provider's own internal network. At each connection location (gateway), the Cloud provider uses BGP to advertise all of the prefixes in the enterprise's VPC, regardless of which Cloud DC a given prefix is actually in. This can result in inefficient routing for the end-to-end data path.

Another roadblock is the lack of a standard way to express and enforce consistent security policies for workloads that not only use virtual addresses, but in which are also very likely hosted in different locations within the Cloud DC [RFC8192]. The current VPN path computation and bandwidth allocation schemes may not be flexible enough to address the need for enterprises to rapidly connect to dynamically instantiated (or removed) workloads and applications regardless of their location/nature (i.e., third party cloud DCs).

#### 7. Problem with using IPsec tunnels to Cloud DCs

As described in the previous section, many Cloud operators expose their gateways for external entities (which can be enterprises themselves) to directly establish IPsec tunnels. Enterprises can also instantiate virtual routers within Cloud DCs to connect to their on-premises devices via IPsec tunnels.

### 7.1. Scaling Issues with IPsec Tunnels

If there is only one enterprise location that needs to reach the Cloud DC, an IPsec tunnel is a very convenient solution.

However, many medium-to-large enterprises have multiple sites and multiple data centers. For multiple sites to communicate with workloads and apps hosted in cloud DCs, Cloud DC gateways have to maintain many IPsec tunnels to all those locations. In addition, each of those IPsec Tunnels requires pair-wise periodic key refreshment. For a company with hundreds or thousands of locations, there could be hundreds (or even thousands) of IPsec tunnels terminating at the cloud DC gateway, which is very processing intensive. That is why many cloud operators only allow a limited number of (IPsec) tunnels & bandwidth to each customer.

Alternatively, you could use a solution like group encryption where a single IPsec SA is necessary at the GW but the drawback is key distribution and maintenance of a key server, etc.

### 7.2. Poor performance over long distance

When enterprise CPEs or gateways are far away from cloud DC gateways or across country/continent boundaries, performance of IPsec tunnels over the public Internet can be problematic and unpredictable. Even though there are many monitoring tools available to measure delay and various performance characteristics of the network, the measurement for paths over the Internet is passive and past measurements may not represent future performance.

Many cloud providers can replicate workloads in different available zones. An App instantiated in a cloud DC closest to clients may have to cooperate with another App (or its mirror image) in another region or database server(s) in the on-premises DC. This kind of coordination requires predictable networking behavior/performance among those locations.

## 8. End-to-End Security Concerns for Data Flows

When IPsec tunnels established from enterprise on-premises CPEs are terminated at the Cloud DC gateway where the workloads or applications are hosted, some enterprises have concerns regarding traffic to/from their workload being exposed to others behind the data center gateway (e.g., exposed to other organizations that have workloads in the same data center).

To ensure that traffic to/from workloads is not exposed to unwanted entities, IPsec tunnels may go all the way to the workload (servers, or VMs) within the DC.

#### 9. Requirements for Dynamic Cloud Data Center VPNs

To address the aforementioned issues, any solution for enterprise VPNs that includes connectivity to dynamic workloads or applications in cloud data centers should satisfy a set of requirements:

- The solution should allow enterprises to take advantage of the current state-of-the-art in VPN technology, in both traditional MPLS-based VPNs and IPsec-based VPNs (or any combination thereof) that run over the public Internet.
- The solution should not require an enterprise to upgrade all their existing CPEs.
- The solution should support scalable IPsec key management among all nodes involved in DC interconnect schemes.
- The solution needs to support easy and fast, on-the-fly, VPN connections to dynamic workloads and applications in third party data centers, and easily allow these workloads to migrate both within a data center and between data centers.
- Allow VPNs to provide bandwidth and other performance guarantees.
- Be a cost-effective solution for enterprises to incorporate dynamic cloud-based applications and workloads into their existing VPN environment.

#### 10. Security Considerations

The draft discusses security requirements as a part of the problem space, particularly in sections 4, 5, and 8.

Solution drafts resulting from this work will address security concerns inherent to the solution(s), including both protocol aspects and the importance (for example) of securing workloads in cloud DCs and the use of secure interconnection mechanisms.

## 11. IANA Considerations

This document requires no IANA actions. RFC Editor: Please remove this section before publication.

## 12. References

### 12.1. Normative References

### 12.2. Informative References

[RFC2735] B. Fox, et al "NHRP Support for Virtual Private networks". Dec. 1999.

[RFC8192] S. Hares, et al "Interface to Network Security Functions (I2NSF) Problem Statement and Use Cases", July 2017

[ITU-T-X1036] ITU-T Recommendation X.1036, "Framework for creation, storage, distribution and enforcement of policies for network security", Nov 2007.

[RFC6071] S. Frankel and S. Krishnan, "IP Security (IPsec) and Internet Key Exchange (IKE) Document Roadmap", Feb 2011.

[RFC4364] E. Rosen and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)", Feb 2006

[RFC4664] L. Andersson and E. Rosen, "Framework for Layer 2 Virtual Private Networks (L2VPNs)", Sept 2006.

## 13. Acknowledgments

Many thanks to Alia Atlas, Chris Bowers, Paul Vixie, Paul Ebersman, Timothy Morizot, Ignas Bagdonas, Michael Huang, Liu Yuan Jiao, Katherine Zhao, and Jim Guichard for the discussion and contributions.

Authors' Addresses

Linda Dunbar  
Futurewei  
Email: Linda.Dunbar@futurewei.com

Andrew G. Malis  
Malis Consulting  
Email: agmalis@gmail.com

Christian Jacquenet  
Orange  
Rennes, 35000  
France  
Email: Christian.jacquenet@orange.com

Mehmet Toy  
Verizon  
One Verizon Way  
Basking Ridge, NJ 07920  
Email: mehmet.toy@verizon.com





Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: May 7, 2020

Z. Li  
S. Peng  
Huawei Technologies  
D. Voyer  
Bell Canada  
C. Xie  
China Telecom  
P. Liu  
China Mobile  
C. Liu  
China Unicom  
K. Ebisawa  
Toyota Motor Corporation  
S. Previdi  
Individual  
J. Guichard  
Futurewei Technologies Ltd.  
November 04, 2019

Application-aware IPv6 Networking (APN6) Framework  
draft-li-apn6-framework-00

Abstract

A multitude of applications are carried over the network, which have varying needs for network bandwidth, latency, jitter, and packet loss, etc. Some new emerging applications (e.g. 5G) have very demanding performance requirements. However, in current networks, the network and applications are decoupled, that is, the network is not aware of the applications' requirements in a fine granularity. Therefore, it is difficult to provide truly fine-granularity traffic operations for the applications and guarantee their SLA requirements.

This document proposes a new framework, named Application-aware IPv6 Networking (APN6), which makes use of IPv6 encapsulation to convey the application characteristic information such as application identification and its network performance requirements into the network to facilitate service provisioning, perform application-level traffic steering and network resource adjustment.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

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

working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

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."

This Internet-Draft will expire on May 7, 2020.

#### Copyright Notice

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 (<https://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.

#### Table of Contents

|  |    |
|--|----|
| 1. Introduction . . . . .                                      | 3  |
| 2. Specification of Requirements . . . . .                     | 3  |
| 3. Terminology . . . . .                                       | 3  |
| 4. APN6 Framework and Key Components . . . . .                 | 4  |
| 5. APN6 Requirements . . . . .                                 | 6  |
| 5.1. Application-aware Information Conveying Requirements . .  | 6  |
| 5.2. Application-aware Information Handling Requirements . . . | 7  |
| 5.2.1. App-aware SLA Guarantee . . . . .                       | 7  |
| 5.2.2. App-aware network slicing . . . . .                     | 8  |
| 5.2.3. App-aware deterministic networking . . . . .            | 8  |
| 5.2.4. App-aware service function chaining . . . . .           | 9  |
| 5.2.5. App-aware network measurement . . . . .                 | 9  |
| 5.3. Security requirements . . . . .                           | 9  |
| 6. IANA Considerations . . . . .                               | 9  |
| 7. Security Considerations . . . . .                           | 10 |
| 8. Acknowledgements . . . . .                                  | 10 |
| 9. Contributors . . . . .                                      | 10 |
| 10. References . . . . .                                       | 10 |
| 10.1. Normative References . . . . .                           | 10 |
| 10.2. Informative References . . . . .                         | 11 |
| Authors' Addresses . . . . .                                   | 11 |

## 1. Introduction

A multitude of applications are carried over the network, which have varying needs for network bandwidth, latency, jitter, and packet loss, etc. Some applications such as online gaming and live video streaming has very demanding network requirements and therefore require special treatment in the network. However, in current networks, the network and applications are decoupled, that is, the network is not aware of the applications' requirements in a fine granularity. Therefore, it is difficult to provide truly fine-granularity traffic operations for the applications and guarantee their SLA requirements accordingly.

[I-D.li-apn6-problem-statement-usecases] describes the challenges of traditional differentiated service provisioning methods, such as five tuples used for ACL/PBR causing coarse granularity, DPI imposing high CAPEX & OPEX and security issues, as well as orchestration and SDN-based solution causing long control loops.

This document proposes a new framework, named Application-aware IPv6 Networking, aiming to guarantee fine-granularity SLA requirements of applications, which make use of IPv6 encapsulation to convey the application characteristic such as application identification and its network performance requirements into the network to determine the path, steer traffic, and perform network resource adjustment.

## 2. Specification of Requirements

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

This document is not a protocol specification and the key words in this document are used for clarity and emphasis of requirements language.

## 3. Terminology

ACL: Access Control List

APN6: Application-aware IPv6 Networking

DPI: Deep Packet Inspection

PBR: Policy Based Routing

QoE: Quality of Experience

#### 4. APN6 Framework and Key Components

The APN6 framework is shown in Figure 1. The key components include Application-aware App, App-aware Edge Device, App-aware-process Head-End, App-aware-process Mid-Point, and App-aware-process End-Point.

Packets carry application characteristic information (i.e. application-aware information) which includes the following information:

- o Application-aware identification information: identifying application, the user of application, i.e. the packets as part of the traffic flow belonging to a specific SLA level/Application/User;
- o Network performance requirements information: specifying at least one of the following parameters: bandwidth, delay, delay variation, packet loss ratio, security, etc.

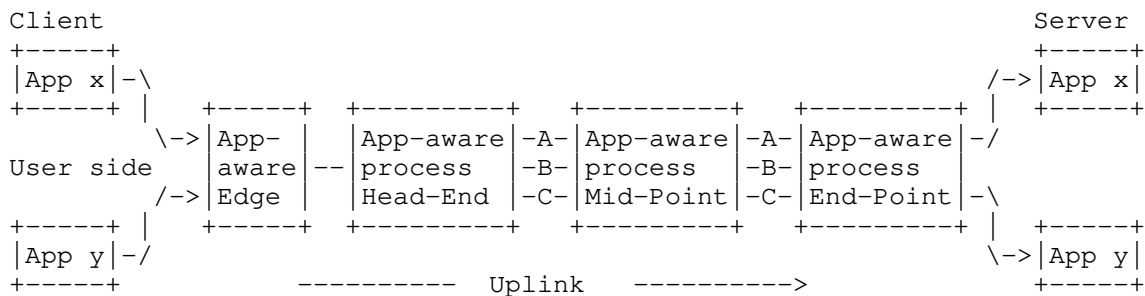


Figure 1 APN6 Framework and Key Components

The key components are introduced as follows.

1. Application-aware App: The host obtains the application characteristic information of the Application-aware App and generates the packets which carry the application characteristic information in IPv6 encapsulation. If carried in the packets, this information is used by the App-aware-process Head-End to determine the path between the App-aware-process Head-End and the App-aware-process End-Point for forwarding the packets to their destination, that is, to steer the packet in to a given policy which satisfies the application requirements. In the APN6 framework, the application is not mandatory to be application-aware.

2. App-aware Edge Device: This network device receives packets from applications and obtains the application characteristic information. If the application is not Application-aware App, the application

characteristic information can be obtained by packet inspection, derived from services information such as double VLAN tagging (C-VLAN and S-VLAN), or added according to the local policies which is out of the scope of this document. The App-aware Edge Device adds the application characteristic information in IPv6 encapsulation on behalf of the application. The packets carrying the application characteristic information will be sent to the App-aware-process Head-End, and the application characteristic information will be used to determine the path between the App-aware-process Head-End and the App-aware-process End-Point for forwarding the packets.

3. App-aware-process Head-End: This network device receives packets and obtains the application characteristic information. A set of paths, tunnels or SR policy, exist between the App-aware-process Head-End and the App-aware-process End-Point. The App-aware-process Head-End maintains the matching relationship between the application characteristic information and the paths between the App-aware-process Head-End and the App-aware-process End-Point. The App-aware-process Head-End determines the path between the App-aware-process Head-End and the App-aware-process End-Point according to the application characteristic information carried in the packets and the matching relationship with it, which satisfies the service requirements of the application. If there is no such matching path found, the App-aware-process Head-End can set up a path towards the App-aware-process End-Point, and the matching relationship will be stored. The App-aware-process Head-End forwards the packets along the path. The application information conveyed by the packet received from the App-aware Edge Device can also be copied or be mapped to the out IPv6 extension header for further application-aware process.

4. App-aware-process Mid-Point: The Mid-Point provides the path service according to the path set up by the App-aware-process Head-End which satisfies the service requirements conveyed by the IPv6 packets. The Mid-Point may also adjust the resource locally to guarantee the service requirements depending on a specific policy and the application-aware information conveyed by the packet. Policy definitions and mechanisms are out of the scope of this document.

5. App-aware-process End-Point: The process of the specific service path will end at the End-Point. The service requirements information can be removed at the End-Point together with the outer IPv6 encapsulation or go on to be conveyed with the IPv6 packets.

In this way the network is aware of the service requirements expressed by the applications explicitly. According to the service requirement information carried in the IPv6 packets the network is able to adjust its resources fast in order to satisfy the service

requirement of applications. The flow-driven method also reduces the challenges of inter-operability and long control loop.

## 5. APN6 Requirements

Utilizing IPv6 encapsulation (e.g. IPv6 header as well as, possibly, extension headers), the application-aware information is conveyed into the network which performs service provisioning, traffic steering, and SLA guarantee according to such information. This section specifies the requirements for supporting the APN6 framework, including the requirements for conveying and handling the application-aware information and related security requirements.

### 5.1. Application-aware Information Conveying Requirements

The application-aware information includes application-aware identification information and network performance requirements information.

1. Application-aware identification information includes the following identifiers (IDs),
  - \* SLA level: indicating the level of SLA requirement of the application such as Gold, Silver, Bronze. In some cases, color (e.g. red, green) can be used to indicate the SLA level.
  - \* Application ID: identifying an application.
  - \* User ID: identifying the user of the application.
  - \* Flow ID: identifying the flow which the application traffic belongs to.

The different combinations of the IDs can be used to provide different granularity of the service provisioning and SLA guarantee for the traffic.

2. Network performance requirements information includes the following parameters,
  - \* Bandwidth: the bandwidth requirement of the application traffic
  - \* Latency: the latency requirement of the application
  - \* Jitter: the jitter requirement of the application

The different combinations of the parameters are for further expressing the more detailed service requirements of an application, conveyed together with the Application-aware identifiers, which can be used to match to appropriate tunnels/SR Policies, queues that can satisfy these service requirements. If not available, new tunnels/SR Policies can also be triggered to be set up.

[REQ 1a]. Application-aware identification information MUST include Application ID to indicate the application that generates the packet.

[REQ 1b]. SLA level is RECOMMENDED to be included in the Application-aware identification information.

[REQ 1c]. User ID and Flow ID are OPTIONAL to be included in the Application-aware identification information.

[REQ 1d]. Network performance requirements information is OPTIONAL.

[REQ 1e]. All the nodes along the path SHOULD be able to process the application-aware information if needed.

[REQ 1f]. The application-aware information can be generated directly by application, or by the application-aware edge devices though packet inspection or local policy.

[REQ 1g]. The application-aware information SHOULD be kept intact when directly copied from the application-aware edge devices and carried in the IPv6 encapsulation.

## 5.2. Application-aware Information Handling Requirements

The app-aware-process Head-End and app-aware-process Mid-Point perform matching operation against the application-aware information, that is, to match IDs and/or service requirements to the corresponding network resources (tunnels/SR policies, queues).

### 5.2.1. App-aware SLA Guarantee

In order to achieve better Quality of Experience (QoE) of end users and engage customers, the network needs to be able to provide fine-granularity and even application-level SLA guarantee [I-D.li-apn6-problem-statement-usecases].

[REQ 2-1a]. With the application-aware information, the App-aware-process Head-End SHOULD be able to steer the traffic to the tunnel/SR policy that satisfies the matching operation.



[REQ 2-1b]. With the application-aware information, the App-aware-process Head-End SHOULD be able to trigger the setup of the tunnel/SR policy that satisfies the matching operation.

[REQ 2-1c]. With the application-aware information, the App-aware-process Head-End and Mid-Point SHOULD be able to steer the traffic to the queue that satisfies the matching operation.

[REQ 2-1d]. With the application-aware information, the App-aware-process Head-End and Mid-Point SHOULD be able to trigger the configuration of the queue that satisfies the matching operation.

#### 5.2.2. App-aware network slicing

Network slicing provides ways to partition the network infrastructure in either control plane or data plane into multiple network slices that are running in parallel. The resources on each node need to be associated to corresponding slices.

[REQ 2-2a]. With the application-aware information, the App-aware-process Head-End SHOULD be able to steer the traffic to the slice that satisfies the matching operation.

[REQ 2-2a]. With the application-aware information, the App-aware-process Mid-Point SHOULD be able to associate the traffic to the resources in the slice that satisfies the matching operation.

#### 5.2.3. App-aware deterministic networking

Along the path each node needs to provide guaranteed bandwidth, bounded latency, and other properties relevant to the transport of time-sensitive data for the Detnet flows that coexist with the best-effort traffic.

[REQ 2-3a]. With the application-aware information, the App-aware-process Head-End SHOULD be able to steer the traffic to the appropriate path that satisfies the matching operation.

[REQ 2-3b]. With the application-aware information, the App-aware-process Head-End SHOULD be able to trigger the setup of the appropriate path that satisfies the matching operation for the Detnet flows.

[REQ 2-3c]. With the application-aware information, the App-aware-process Mid-Point SHOULD be able to associate the traffic to the resources along the path that satisfies the performance guarantee.

[REQ 2-3d]. With the application-aware information, the App-aware-process Mid-Point SHOULD be able to reserve the resources for the Detnet flows along the path that satisfies the performance guarantee.

#### 5.2.4. App-aware service function chaining

The end-to-end service delivery often needs to go through various service functions, including traditional network service functions such as firewalls, DPI as well as new application-specific functions, both physical and virtual. SFC is applicable to both fixed and mobile networks as well as data center networks.

[REQ 2-4a]. With the application-aware information, the App-aware-process devices SHOULD be able to steer the traffic to the appropriate service function.

[REQ 2-4b]. The App-aware-process devices SHOULD be able to process the application-aware information carried in the packets.

#### 5.2.5. App-aware network measurement

Network measurement can be used for locating silent failure and predicting QoE satisfaction, which enables real-time SLA awareness/proactive OAM.

[REQ 2-5a]. With the application-aware identification information, the App-aware-process devices SHOULD be able to perform IOAM based on the Application ID.

[REQ 2-5a]. With the application-aware information, the network measurement results can be reported based on the Application ID and verify whether the performance requirements of the application are satisfied.

#### 5.3. Security requirements

[REQ 3a]. The security mechanism defined for APN6 MUST allow an operator to prevent applications sending arbitrary application-aware information without agreement with the operator.

[REQ 3b]. The security mechanism defined for APN6 MUST prevent an application requesting a service that is not entitled to get.

#### 6. IANA Considerations

This document does not include an IANA request.

## 7. Security Considerations

[I-D.li-apn6-problem-statement-usecases] and section 5.3 describe the security considerations and requirements for APN6.

## 8. Acknowledgements

The authors would like to acknowledge Robert Raszuk (Bloomberg LP) and Yukito Ueno (NTT Communications Corporation) for their valuable reviews and comments.

## 9. Contributors

Liang Geng  
China Mobile  
China

Email: gengliang@chinamobile.com

Chang Cao  
China Unicom  
China

Email: caoc15@chinaunicom.cn

Cong Li  
China Telecom  
China

Email: licong.bri@chinatelecom.cn

## 10. References

### 10.1. Normative References

- [I-D.li-apn6-problem-statement-usecases]  
Li, Z., Peng, S., Voyer, D., Xie, C., Liu, P., Liu, C.,  
Ebisawa, K., Ueno, Y., Previdi, S., and J. Guichard,  
"Problem statement and use cases of Application-aware IPv6  
Networking (APN6)", draft-li-apn6-problem-statement-  
usecases-00 (work in progress), September 2019.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate  
Requirement Levels", BCP 14, RFC 2119,  
DOI 10.17487/RFC2119, March 1997,  
<<https://www.rfc-editor.org/info/rfc2119>>.

- [RFC7665] Halpern, J., Ed. and C. Pignataro, Ed., "Service Function Chaining (SFC) Architecture", RFC 7665, DOI 10.17487/RFC7665, October 2015, <<https://www.rfc-editor.org/info/rfc7665>>.
- [RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <<https://www.rfc-editor.org/info/rfc8200>>.
- [RFC8578] Grossman, E., Ed., "Deterministic Networking Use Cases", RFC 8578, DOI 10.17487/RFC8578, May 2019, <<https://www.rfc-editor.org/info/rfc8578>>.

## 10.2. Informative References

- [RFC3272] Awduche, D., Chiu, A., Elwalid, A., Widjaja, I., and X. Xiao, "Overview and Principles of Internet Traffic Engineering", RFC 3272, DOI 10.17487/RFC3272, May 2002, <<https://www.rfc-editor.org/info/rfc3272>>.

## Authors' Addresses

Zhenbin Li  
Huawei Technologies  
China

Email: [lizhenbin@huawei.com](mailto:lizhenbin@huawei.com)

Shuping Peng  
Huawei Technologies  
China

Email: [pengshuping@huawei.com](mailto:pengshuping@huawei.com)

Daniel Voyer  
Bell Canada  
Canada

Email: [daniel.voyer@bell.ca](mailto:daniel.voyer@bell.ca)

Chongfeng Xie  
China Telecom  
China

Email: xiechf.bri@chinatelecom.cn

Peng Liu  
China Mobile  
China

Email: liupengyjy@chinamobile.com

Chang Liu  
China Unicom  
China

Email: liuc131@chinaunicom.cn

Kentaro Ebisawa  
Toyota Motor Corporation  
Japan

Email: ebisawa@toyota-tokyo.tech

Stefano Previdi  
Individual  
Italy

Email: stefano@previdi.net

James N Guichard  
Futurewei Technologies Ltd.  
USA

Email: jguichar@futurewei.com

Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: May 7, 2020

Z. Li  
S. Peng  
Huawei Technologies  
D. Voyer  
Bell Canada  
C. Xie  
China Telecom  
P. Liu  
China Mobile  
C. Liu  
China Unicom  
K. Ebisawa  
Toyota Motor Corporation  
S. Previdi  
Individual  
J. Guichard  
Futurewei Technologies Ltd.  
November 04, 2019

Problem Statement and Use Cases of Application-aware IPv6 Networking  
(APN6)  
draft-li-apn6-problem-statement-usecases-01

Abstract

Network operators are facing the challenge of providing better network services for users. As the ever developing 5G and industrial verticals evolve, more and more services that have diverse network requirements such as ultra-low latency and high reliability are emerging, and therefore differentiated service treatment is desired by users. However, network operators are typically unaware of which applications are traversing their network infrastructure, which means that only coarse-grained services can be provided to users. As a result, network operators are only evolving their infrastructure to be large but dumb pipes without corresponding revenue increases that might be enabled by differentiated service treatment. As network technologies evolve including deployments of IPv6 and SRv6, the programmability provided by IPv6 and SRv6 encapsulations can be augmented by conveying application related information into the network. Adding application knowledge to the network layer allows applications to specify finer granularity requirements to the network operator.

This document analyzes the existing problems caused by lack of application awareness, and outlines various use cases that could benefit from an Application-aware IPv6 Networking (APN6) architecture.

## Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

## Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

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."

This Internet-Draft will expire on May 7, 2020.

## Copyright Notice

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 (<https://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.

## Table of Contents

|  |   |
|--|---|
| 1. Introduction . . . . .  | 3 |
| 2. Terminology . . . . .   | 3 |
| 3. Problem Statement . . . . .   | 4 |
| 3.1. Large but Dumb Pipe . . . . .   | 4 |
| 3.2. Network on Its Own . . . . .  | 4 |
| 3.3. Decoupling of Network and Applications . . . . .                        | 4 |
| 3.4. Challenges of Traditional Differentiated Service Provisioning . . . . . | 5 |

|  |    |
|--|----|
| 3.5. Challenges of Supporting New 5G and Edge Computing Technologies . . . . . | 6  |
| 4. Key Elements of Application-aware IPv6 Networking (APN6) . . . . .          | 6  |
| 5. Use cases for Application-aware IPv6 Networking (APN6) . . . . .            | 8  |
| 5.1. Application-aware SLA Guarantee . . . . .                                 | 8  |
| 5.2. Application-aware network slicing . . . . .                               | 9  |
| 5.3. Application-aware Deterministic Networking . . . . .                      | 9  |
| 5.4. Application-aware Service Function Chaining . . . . .                     | 10 |
| 5.5. Application-aware Network Measurement . . . . .                           | 10 |
| 6. IANA Considerations . . . . .   | 11 |
| 7. Security Considerations . . . . .   | 11 |
| 8. Acknowledgements . . . . .  | 11 |
| 9. Contributors . . . . .  | 11 |
| 10. References . . . . .   | 12 |
| 10.1. Normative References . . . . .   | 12 |
| 10.2. Informative References . . . . .   | 12 |
| Authors' Addresses . . . . .   | 12 |

## 1. Introduction

Due to the requirement for differentiated traffic treatment driven by diverse new services, the ability to convey the characteristics of an application's traffic flow and program the network infrastructure accordingly to provide fine-grained service assurance is becoming increasingly necessary for network operators. The Application-aware IPv6 Networking (APN6) architecture is being defined to address the requirements and use cases described in this document. APN6 takes advantage of network programmability by conveying application related information in the data plane allowing applications to specify finer grained requirements to the network infrastructure.

## 2. Terminology

ACL: Access Control List

APN6: Application-aware IPv6 Networking

DPI: Deep Packet Inspection

PBR: Policy Based Routing

QoE: Quality of Experience

SDN: Software Defined Networking



### 3. Problem Statement

This section summarizes the challenges currently faced by network operators when attempting to provide fine-grained traffic operations to satisfy the various application-awareness requirements demanded by new services that require differentiated service treatment.

#### 3.1. Large but Dumb Pipe

In today's networks, the infrastructure through which user traffic is forwarded is not able to determine information about the packet, including which application the traffic belongs to, without the introduction of middleware such as DPI, that is, the network and applications are decoupled. It is therefore difficult for network operators to provide fine-grained traffic operations for performance-demanding applications. In order to satisfy the SLA requirements network operators continue to increase the network bandwidth but only carrying very light traffic load (around 30%-40% of its capacity). This situation greatly increases the CAPEX and OPEX but only brings very little revenue from the carried services.

#### 3.2. Network on Its Own

As the network evolves, technologies such as VPN/TE/FRR play important roles in satisfying service isolation, SLA guarantee, and high reliability, etc. These network technologies have themselves been evolving, introducing new features that forces the network operator to be continuously upgrading their network infrastructure. However, none of these network technologies make the network aware of which application traffic belongs to and the fine granularity requirements of the application. Therefore, such continuous network infrastructure upgrade doesn't always enable true fine-grained traffic operation, therefore reducing the ability to bring corresponding revenue increase.

#### 3.3. Decoupling of Network and Applications

MPLS played a very important role in helping the network enter the generation of All-IP successfully. However, MPLS doesn't allow a close interworking with the application layer since MPLS encapsulation is, typically, not used by the packet source.

As new services continuously evolve, more encapsulations are required, and this isolation and decoupling has further become the blockage towards the seamless convergence of the network and applications.

### 3.4. Challenges of Traditional Differentiated Service Provisioning

Several IETF activities have been reviewed which are primarily intended to evolve the IP architecture to support new service definitions which allow preferential or differentiated treatment to be accorded to certain types of traffic. The challenge when using traditional ways to guarantee an SLA is that the packets are not able to carry enough information for indicating applications and expressing their service/SLA requirements. The network devices mainly rely on the 5-tuple of the packets or DPI. However, there are some challenges for these traditional methods in differentiated service provisioning:

#### 1. Five Tuples used for ACL/PBR

Five tuples are widely used for ACL/PBR matching of traffic. However, these features cannot provide enough information for the fine-grained service process, and can only provide indirect application information which needs to be translated in order to indicate a specific application.

#### 2. Deep Packet Inspection (DPI)

If more information is needed, it must be extracted using DPI which can inspect deep into the packets for application specific information. However, this will introduce more CAPEX and OPEX for the network operator and imposes security challenges.

#### 3. Orchestration and SDN-based Solution

In the era of SDN, typically, an SDN controller is used to manage and operate the network infrastructure and orchestrator elements introduce application requirements so that the network is programmed accordingly. The SDN controller can be aware of the service requirements of the applications on the network through the interface with the orchestrator, and the service requirement is used by the controller for traffic management over the network. However, this method raises the following problems:

1) The whole loop is long and time-consuming which is not suitable for fast service provisioning for critical applications;

2) Too many interfaces are involved in the loop, as shown in Figure 1, which introduce challenges of standardization and interoperability.

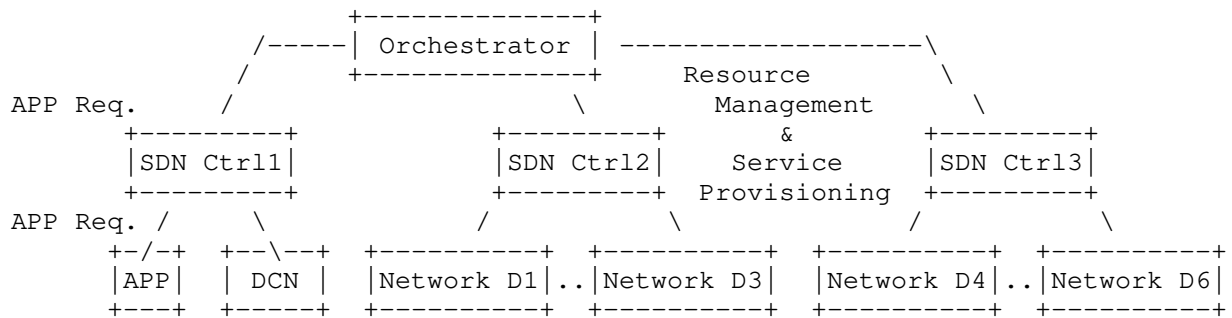


Figure 1. Many interfaces involved in the long service-provisioning loop

### 3.5. Challenges of Supporting New 5G and Edge Computing Technologies

New technologies such as 5G, IoT, and edge computing, are continuously developing leading to more and more new types of services accessing the network. Large volumes of network traffic with diverse requirements such as low latency and high reliability are therefore rapidly increasing. If traditional methods for differentiation of traffic continue to be utilized, it will cause much higher CAPEX and OPEX to satisfy the ever-developing applications' diverse requirements.

### 4. Key Elements of Application-aware IPv6 Networking (APN6)

Application-aware IPv6 Networking (APN6) aims to address the aforementioned problems associated with fine-grained traffic operations that are required in order to satisfy the various application-awareness requirements demanded by new services that need differentiated service treatment. APN6 conveys information into the network infrastructure about the characteristics of the application associated with a traffic flow (including application identification and network performance requirements), allowing the network to quickly adapt and perform the necessary network resource adjustments to maintain SLA performance guarantees, and hence better serve application fine-grained service requirements.

The advantages of using IPv6 to support APN6 include,

1. **Simplicity:** Conveying application information with IPv6 encapsulation can just be based on IP reachability.
2. **Seamless convergence:** Much easier to achieve seamless convergence between applications and network since both are based on IPv6.

3. Great extensibility: IPv6 encapsulation including its extension headers can be used to carry very rich information relevant to applications.
4. Good compatibility: On-demand network upgrade and service provisioning. If the application information is not recognized by the node, the packet will be forwarded based on pure IPv6, which ensure backward compatibility.
5. Little dependency: Information conveying and service provisioning are only based on the forwarding plane of devices, which is different from the Orchestration and SDN-based solution which involves multiple elements and diverse interfaces.
6. Quick response: Flow-driven and direct response from devices since it is based on the forwarding plane.

APN6 has the following key elements:

1. Application information should be conveyed in the data plane through augmentation of existing encapsulations such as IPv6 and/or SRv6. The conveyed application characteristic information (application-aware information) includes application identification and/or its network performance requirements. This element should not be enforced but provide an open option for applications to decide whether to input this application-aware information into their data stream.
2. Application information and network service provisioning matching providing fine-granularity network service provisioning (traffic operations) and SLA guarantee based on the application-aware information carried in APN6 packets. This element provides the network capabilities to applications. According to the application-aware information, appropriate network services are selected, provisioned, and provided to the demanding applications to satisfy their performance requirements.
3. Network measurement of network performance and update the match between the applications and corresponding network services for better fine-granularity SLA compliance. The network measurement methods include in-band and out-of-band, passive, active, per-packet, per-flow, per node, end-to-end, etc. These methods can also be integrated.

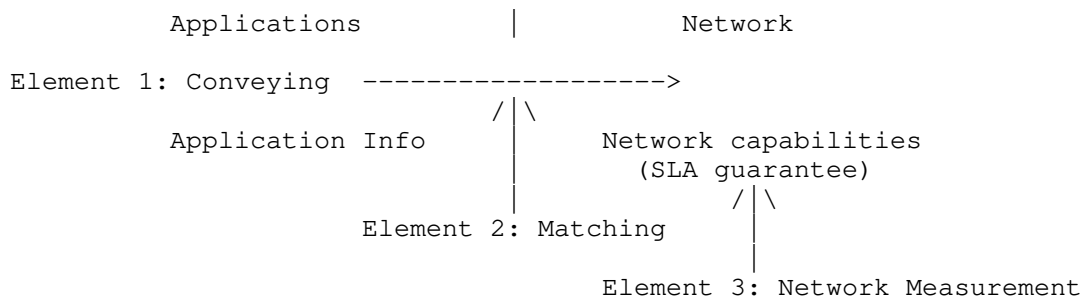


Figure 2. Illustration of the key elements of APN6

## 5. Use cases for Application-aware IPv6 Networking (APN6)

This section provides the use cases that can benefit from the application awareness introduced by APN6. The corresponding requirements for APN6 are also outlined.

### 5.1. Application-aware SLA Guarantee

One of the key objectives of APN6 is for network operators to provide fine-granularity SLA guarantees instead of coarse-grain traffic operations. Among various applications being carried and running in the network, some revenue-producing applications such as online gaming, video streaming, and enterprise video conferencing have much more demanding performance requirements such as low network latency and high bandwidth. In order to achieve better Quality of Experience (QoE) for end users and engage customers, the network needs to be able to provide fine-granularity and even application-level SLA guarantee. Differentiated service provisioning is also desired.

One of the key objective of APN6 is for network operators to provide fine-granularity SLA guarantees instead of coarse-grain traffic operations. This will enable them to provide differentiated services for different applications and increase revenue accordingly.

The APN6 architecture design MUST address the following requirements:

- o APN6 needs to perform the three key elements as described in Section 4.
- o Support application-level fine-granularity traffic operation that may include finer QoS scheduling.

## 5.2. Application-aware network slicing

More and more applications/services with diverse requirements are being carried over and sharing the network operators' network infrastructure. However, it is still desirable to have customized network transport that can support some application's specific requirements, taking into consideration service and resource isolation, which drives the concept of network slicing.

Network slicing provides ways to partition the network infrastructure in either the control plane or data plane into multiple network slices that are running in parallel. These network slices can serve diverse services and fulfill their various requirements at the same time. For example, the mission critical application that requires ultra-low latency and high reliability can be provisioned over a separate network slice.

The APN6 architecture design MUST address the following requirements:

- o APN6 needs to perform the three key elements as described in Section 4 in the context of network slicing. To be more specific, for element 2, it needs to match to a specific network slice according to the application information carried in the APN6 packets. The network measurement in element 3 also needs to happen within each network slice.

## 5.3. Application-aware Deterministic Networking

[RFC8578] documents use cases for diverse industry applications that require deterministic flows over multi-hop paths. Deterministic flows provide guaranteed bandwidth, bounded latency, and other properties relevant to the transport of time-sensitive data, and can coexist on an IP network with best-effort traffic. It also provides for highly reliable flows through provision for redundant paths.

The APN6 architecture design MUST address the following requirements:

- o APN6 needs to perform the three key elements as described in Section 4 in the context of deterministic networking. To be more specific, for the element 2, it needs to match to a specific deterministic path according to the application information carried in the APN6 packets. The network measurement in element 3 also needs to be performed on each application-aware deterministic path.

#### 5.4. Application-aware Service Function Chaining

End-to-end service delivery often needs to go through various service functions, including traditional network service functions such as firewalls, DPIs as well as new application-specific functions, both physical and virtual. The definition and instantiation of an ordered set of service functions and subsequent steering of the traffic through them is called Service Function Chaining (SFC) [RFC7665]. SFC is applicable to both fixed and mobile networks as well as data center networks.

Generally, in order to manipulate a specific application traffic along the SFC, a DPI needs to be deployed as the first service function of the chain to detect the application, which will impose high CAPEX and consume long processing times. For encrypted traffic, it even becomes impossible to inspect the application.

The APN6 architecture design MUST address the following requirements:

- o APN6 needs to perform the three key elements as described in Section 4 in the context of service function chaining. To be more specific, for element 1 class information can be conveyed. For element 2, it needs to match to a specific service function chain and subsequent steering according to the application information carried in the APN6 packets. The network measurement in element 3 also needs to happen within each app-aware service function chain.

#### 5.5. Application-aware Network Measurement

Network measurement can be used for locating silent failure and predicting QoE satisfaction, which enables real-time SLA awareness/proactive OAM. Operations, Administration, and Maintenance (OAM) refers to a toolset for fault detection and isolation, and network performance measurement. In-situ Operations, Administration, and Maintenance (IOAM) records operational and telemetry information in the packet while the packet traverses a path between two points in the network.

The APN6 architecture MUST address the following requirements:

- o APN6 needs to perform the two key elements as described in Section 4 in the context of network measurement. The network measurement in element 3 does not need to be considered here.

## 6. IANA Considerations

This document does not include an IANA request.

## 7. Security Considerations

Since the application information is conveyed into the network, it does involve some security and privacy issues.

First, APN6 only provides the capability to the applications to provide their profiles and requirements to the network, but it leaves the applications to decide whether to input this information. If the applications decide not to provide any information, they will be treated in the same way as today's network and cannot get the benefits from APN6.

Once the application information has been carried in the IPv6 packets and conveyed into the network, the IPv6 extension headers, AH and ESP, can be used to guarantee the authenticity of the added application information.

Any scheme involving an information exchange between layers (application and network layers in this case) will obviously require an accurate valuation of security mechanism in order to prevent any leak of critical information. Some additional considerations may be required for multi-domain use cases. For example, how to agree upon which application information/ID to use and guarantee authenticity for packets traveling through multiple domains (network operators).

## 8. Acknowledgements

The authors would like to acknowledge Robert Raszuk (Bloomberg LP) and Yukito Ueno (NTT Communications Corporation) for their valuable review and comments.

## 9. Contributors

Liang Geng  
China Mobile  
China

Email: gengliang@chinamobile.com

Chang Cao  
China Unicom  
China

Email: caoc15@chinaunicom.cn



Cong Li  
China Telecom  
China

Email: licong.bri@chinatelecom.cn

## 10. References

### 10.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC7665] Halpern, J., Ed. and C. Pignataro, Ed., "Service Function Chaining (SFC) Architecture", RFC 7665, DOI 10.17487/RFC7665, October 2015, <<https://www.rfc-editor.org/info/rfc7665>>.
- [RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <<https://www.rfc-editor.org/info/rfc8200>>.
- [RFC8578] Grossman, E., Ed., "Deterministic Networking Use Cases", RFC 8578, DOI 10.17487/RFC8578, May 2019, <<https://www.rfc-editor.org/info/rfc8578>>.

### 10.2. Informative References

- [I-D.ietf-6man-segment-routing-header]  
Filsfils, C., Dukes, D., Previdi, S., Leddy, J., Matsushima, S., and d. daniel.voyer@bell.ca, "IPv6 Segment Routing Header (SRH)", draft-ietf-6man-segment-routing-header-26 (work in progress), October 2019.
- [I-D.ietf-spring-srv6-network-programming]  
Filsfils, C., Camarillo, P., Leddy, J., daniel.voyer@bell.ca, d., Matsushima, S., and Z. Li, "SRv6 Network Programming", draft-ietf-spring-srv6-network-programming-05 (work in progress), October 2019.

## Authors' Addresses

Zhenbin Li  
Huawei Technologies  
China

Email: lizhenbin@huawei.com

Shuping Peng  
Huawei Technologies  
China

Email: pengshuping@huawei.com

Daniel Voyer  
Bell Canada  
Canada

Email: daniel.voyer@bell.ca

Chongfeng Xie  
China Telecom  
China

Email: xiechf.bri@chinatelecom.cn

Peng Liu  
China Mobile  
China

Email: liupengyjy@chinamobile.com

Chang Liu  
China Unicom  
China

Email: liuc131@chinaunicom.cn

Kentaro Ebisawa  
Toyota Motor Corporation  
Japan

Email: ebisawa@toyota-tokyo.tech

Stefano Previdi  
Individual  
Italy

Email: stefano@previdi.net

James N Guichard  
Futurewei Technologies Ltd.  
USA

Email: jguichar@futurewei.com

BFD Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: October 16, 2020

G. Mirsky  
X. Min  
ZTE Corp.  
April 14, 2020

Extended Bidirectional Forwarding Detection  
draft-mirmin-bfd-extended-03

Abstract

This document describes a mechanism to extend the capabilities of Bidirectional Forwarding Detection (BFD). These extensions enable BFD to measure performance metrics like packet loss and packet delay. Also, a method to perform lightweight on-demand authentication is defined in this specification.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

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."

This Internet-Draft will expire on October 16, 2020.

Copyright Notice

Copyright (c) 2020 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 (<https://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.

## Table of Contents

|  |    |
|--|----|
| 1. Introduction . . . . .  | 2  |
| 2. Conventions used in this document . . . . .                           | 3  |
| 2.1. Terminology . . . . .   | 3  |
| 2.2. Requirements Language . . . . .                                     | 3  |
| 3. Extended BFD Control Message . . . . .                                | 3  |
| 3.1. Extended BFD Capability Negotiation . . . . .                       | 5  |
| 3.2. Padding TLV . . . . .   | 6  |
| 3.3. Diagnostic TLV . . . . .  | 7  |
| 3.4. Performance Measurement with Extended BFD Control Message . . . . . | 8  |
| 3.5. Lightweight Authentication . . . . .                                | 9  |
| 3.5.1. Lightweight Authentication Mode Negotiation . . . . .             | 10 |
| 3.5.2. Using Lightweight Authentication . . . . .                        | 11 |
| 4. IANA Considerations . . . . .   | 12 |
| 4.1. Extended BFD Message Types . . . . .                                | 12 |
| 4.2. Lightweight Authentication Modes . . . . .                          | 13 |
| 4.3. Return Codes . . . . .  | 14 |
| 5. Security Considerations . . . . .                                     | 14 |
| 6. References . . . . .  | 15 |
| 6.1. Normative References . . . . .                                      | 15 |
| 6.2. Informative References . . . . .                                    | 15 |
| Appendix A. Acknowledgements . . . . .                                   | 15 |
| Authors' Addresses . . . . .   | 16 |

## 1. Introduction

[RFC5880] has provided the base specification of Bidirectional Detection (BFD) as the light-weight mechanism to monitor a path continuity between two systems and detect a failure in the data-plane. Since its introduction, BFD has been broadly deployed. There were several attempts to introduce new capabilities in the protocol, some more successful than others. One of the significant obstacles to extending BFD capabilities may be seen in the compact format of the BFD control message. This document introduces an Extended BFD control message and describes the use of the new format for new BFD capabilities.

The Extended BFD protocol may be seen as the Operations, Administration, and Maintenance (OAM) protocol that provides both Fault Management (FM) Performance Monitoring (PM) OAM functions. In some networks, for example in a Deterministic Networking (DetNet) domain [RFC8655], it is easier to ensure that a test packet of a single OAM protocol is fate-sharing with data packets rather than map several FM and PM OAM protocols to that DetNet data flow.

## 2. Conventions used in this document

### 2.1. Terminology

BFD: Bidirectional Forwarding Detection

G-ACh Generic Associated Channel

HMAC Hashed Message Authentication Code

MTU Maximum Transmission Unit

PMTUD Path MTU Discovery

PMTUM Path MTU Monitoring

p2p: Point-to-Point

TLV Type, Length, Value

OAM Operations, Administration, and Maintenance

FM Fault Management

PM Performance Monitoring

DetNet Deterministic Networking

### 2.2. Requirements Language

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.

## 3. Extended BFD Control Message

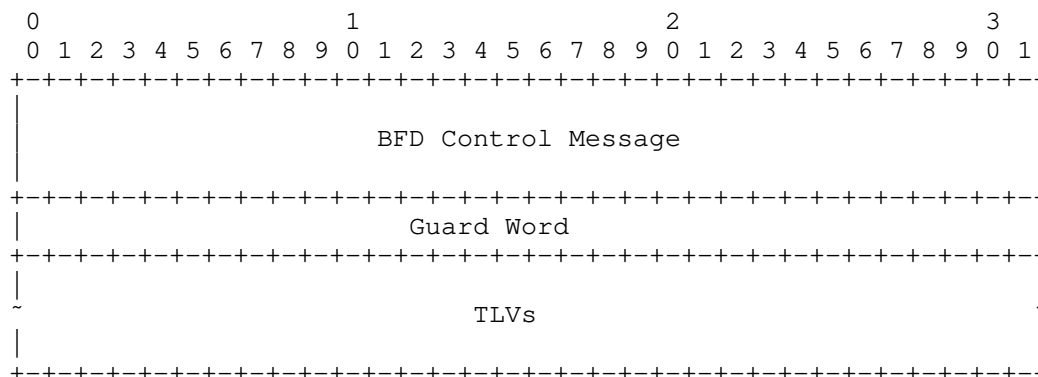


Figure 1: Extended BFD Control Message Format

where fields are defined as the following:

- o BFD control message as defined [RFC5880].
- o Guard word - four octets long field to identify the role of the BFD system - sender or responder.
- o TLVs - variable-length field that contains commands and/or data encoded as type-length-value (TLV).

If an Extended BFD control message is encapsulated in IP/UDP, the value of the Total Length in the IP header includes the length of the Extended BFD control message while the value of the Length field of the BFD control message equals the value as defined in [RFC5880]. If an Extended BFD control message is to be used over Generic Associated Channel (G-ACh), e.g., [RFC6428] new code point for G-ACh may be allocated.

Figure 2 displays the generic TLV format.

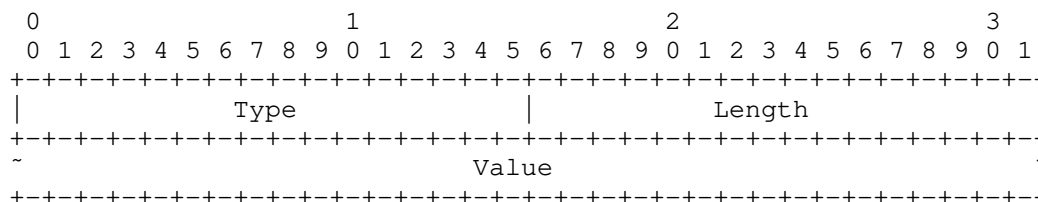


Figure 2: General Type-Length-Value Encoding

where fields are defined as the following:

- o Type - two octets long field that defines the encoding of the Value field
- o Length - two octets long field equals length on the Value field in octets.
- o Value - depends on the Type.

TLVs may be included within other TLVs, in which case the former TLVs are referred to as sub-TLVs. Sub-TLVs have independent types.

### 3.1. Extended BFD Capability Negotiation

A BFD system also referred to as a node in this document, that supports Extended BFD first MUST discover whether other nodes in the given BFD session support the Extended BFD. The node MUST send Extended BFD control message initiating the Poll Sequence as defined in [RFC5880]. If the remote system fails to respond with the Extended BFD control message and the Final flag set, then the initiator node MUST conclude that the BFD peer does not support the use of the Extended BFD control messages.

The first Extended BFD control message initiating the Poll Sequence SHOULD include the Capability TLV that lists capabilities that may be used at some time during the lifetime of the BFD session. The format of the Capability TLV and the capabilities that use the Extended BFD control message are presented in Figure 3.

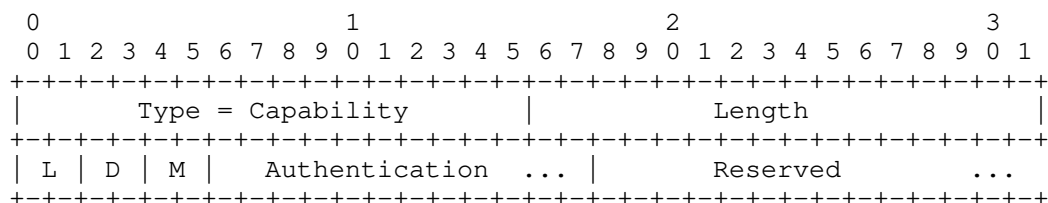


Figure 3: Capability TLV Format

where fields are defined as the following:

- o Type - TBA1 allocated by IANA in Section 4
- o Length - two octets long field equals length on the Capability field in octets. The value of the Length field MUST be a multiple of 4.
- o Loss - two bits size field. The least significant of two bits is set if the node is capable of measuring packet loss using



periodically transmitted Extended BFD control message. The most significant of two bits is set if the node is capable of measuring packet loss using the Poll Sequence with Extended BFD control message.

- o Delay - two bits size field. The least significant of two bits is set if the node is capable of measuring packet delay using periodically transmitted Extended BFD control message. The most significant of two bits is set if the node is capable of measuring packet delay using the Poll Sequence with Extended BFD control message.
- o MTU - two bits size field. Set if the node is capable of using the Extended BFD control message in Path MTU Discovery (PMTUD). or PMTU Monitoring (PMTUM). The least significant of two bits is set if the node is capable of PMTUD/PMTUM using periodically transmitted Extended BFD control message. The most significant of two bits is set if the node is capable of PMTUD/PMTUM using the Poll Sequence with Extended BFD control message.
- o (Lightweight) Authentication - variable-length field. The Authentication field is used by a BFD system to advertise its lightweight authentication capabilities. The format and the use of the Authentication field are defined in Section 3.5.1.
- o Reserved - MUST be zeroed on transmission and ignored on receipt. The Reserved field is zero-padded to align the length of the Capability TLV to a 4-octet boundary.

The remote BFD node that supports this specification MUST respond to the Capability TLV with the Extended BFD control message that includes the Capability TLV listing capabilities the responder supports. The responder MUST set the Final flag in the Extended BFD control message.

### 3.2. Padding TLV

Padding TLV MAY be used to generate Extended BFD control packets of the desired length.

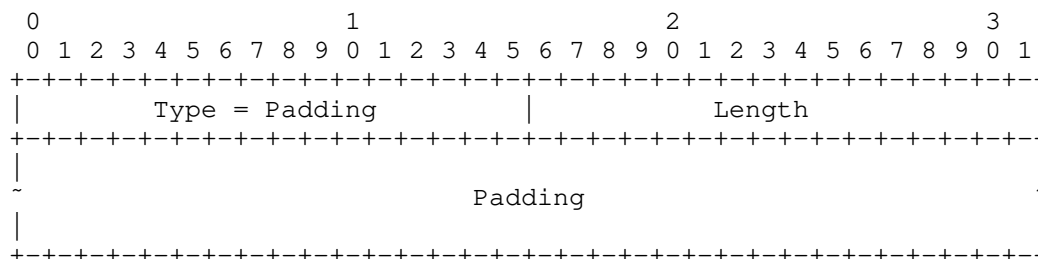


Figure 4: Padding TLV Format

where fields are defined as the following:

- o Type - TBA1 allocated by IANA in Section 4
- o Length - two octets long field equals length on the Padding field in octets.
- o Padding - variable-length field. MUST be zeroed on transmit and ignored on receipt.

Padding TLV MAY be used to generate Extended BFD Control packets of different lengths. That capability is necessary to perform PMTUD, PMTUM, and measure synthetic packet loss and/or packet delay. When Padding TLV is used in combination with one of performance measurement messages carried in Performance Metric TLVs as defined in Section 3.4, Padding TLV MUST follow the Performance Metric TLV.

Padding TLV MAY be used in PMTUM as part of periodically sent Extended BFD Control messages. In this case, the number of consecutive messages that include Padding TLV MUST be not lesser than Detect Multiplier to ensure that the remote BFD peer will detect loss of messages with the Padding TLV. Also, Padding TLV MAY be present in an Extended BFD Control message with the Poll flag set. If the remote BFD peer that supports this specification receives an Extended BFD Control message with Padding TLV, it MUST include the Padding TLV with the Padding field of the same length as in the received packet and set the Final flag.

### 3.3. Diagnostic TLV

Diagnostic TLV MAY be used to characterize the result of the last requested operation.

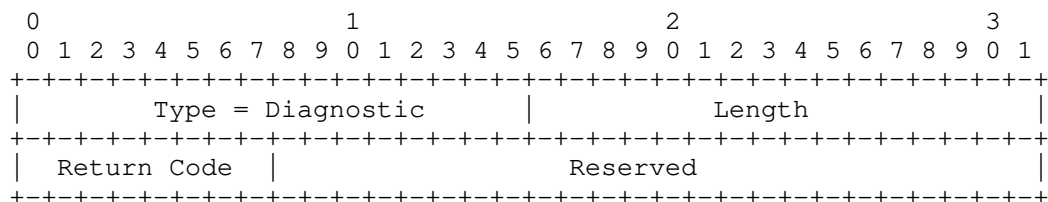


Figure 5: Diagnostic TLV Format

where fields are defined as the following:

- o Type - TBA6 allocated by IANA in Section 4.
- o Length - MUST be set to four.
- o Return Code - eight bits-long field. The responding BFD system can set it to one of the values defined in Section 4.3.
- o Reserved - 24 bits-long field. MUST be zeroed on transmit and ignored on receipt.

### 3.4. Performance Measurement with Extended BFD Control Message

Loss measurement, delay measurement, and loss/delay measurement messages can be used in the Extended BFD control message to support one-way and round-trip measurements. All the messages are encapsulated as TLVs with Type values allocated by IANA, Section 4.

The sender MAY use the Performance Metric TLV (presented in Figure 6) to measure performance metrics and obtain the measurement report from the receiver.

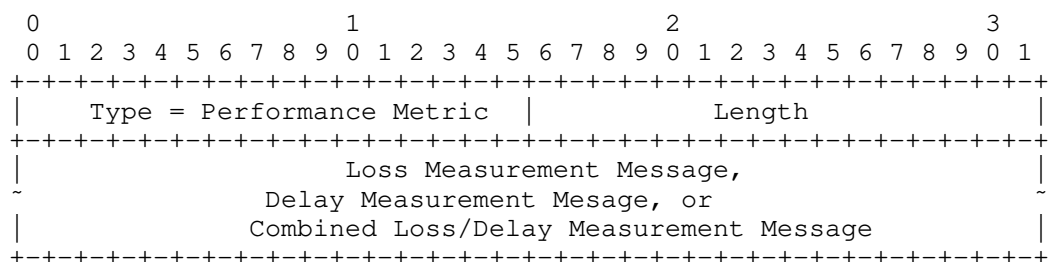


Figure 6: Performance Metric TLV Format

where fields are defined as the following:

- o Type - TBA3 through TBA5 allocated by IANA in Section 4 as follows:
  - \* TBA3 - Loss Measurement Type;
  - \* TBA4 - Delay Measurement Type;
  - \* TBA5 - Combined Loss/Delay Measurement Type
- o Length - two octets long field equals length on the Metric sub-TLVs field in octets. The value of the Length field MUST be a multiple of 4.
- o Value - various performance metrics measured either directly or using synthetic methods accordingly using the messages defined in Sections 3.1 through 3.3 [RFC6374].

To perform one-way loss and/or delay measurement, the BFD node MAY periodically transmit the Extended BFD message with one of the TLVs listed above in Asynchronous mode. To perform synthetic loss measurement, the sender MUST monotonically increment the counter of transmitted test packets. When using Performance Metric TLV for synthetic measurement an Extended BFD Control message MAY also include Padding TLV. In that case, the Padding TLV MUST immediately follow Performance Metric TLV. Also, direct-mode loss measurement, as described in [RFC6374], is supported. Procedures to negotiate and manipulate transmission intervals defined in Sections 6.8.2 and 6.8.3 in [RFC5880] SHOULD be used to control the performance impact of using the Extended BFD for performance measurement in the particular BFD session.

To measure the round-trip loss and/or delay metrics the BFD node transmits the Extended BFD control message with the Performance Metric TLV with the Poll flag set. Before the transmission of the Extended BFD control message with the Performance Metric TLV, the receiver MUST clear the Poll flag and set the Final flag.

### 3.5. Lightweight Authentication

Using BFD without any security measures, for example, by exchanging BFD control packets without authentication, increases the risk of an attack, especially over multiple nodes. Thus, using BFD without security measures may cause false positive as well as false negative defect detection situations. In the former, an attacker may spoof BFD control packets pretending to be a remote peer and can thus view the BFD session operation even though the real path had failed. In the latter, the attacker may spoof altered BFD control message

indicating that the BFD session is un-operational even though the path and the remote BFD peer operate normally.

BFD technology[RFC5880] includes optional authentication protection of BFD control packets to minimize the chances of attacks in a networking system. However, at least some of the supported authentication protocols do not provide sufficient protection in modern networks. Also, current BFD technology requires authentication of each and every BFD control packet. Such an authentication requirement can put a computational burden on networking devices, especially in the Asynchronous mode, at least because authenticating each BFD control packet can require substantial computing resources to support packet exchange at high rates.

This specification defines a lightweight on-demand mode of authentication for a BFD session. The lightweight authentication is an optional mode that can be used when the BFD Authentication [RFC5880] is not in use (bfd.AuthType is zero). The mechanism includes negotiation (Section 3.5.1) and on-demand authentication (Section 3.5.2) phases. During the former, BFD peers advertise supported authentication capabilities and independently select the commonly supported mode of the authentication. In the authentication phase, each BFD system transmits, at certain events and periodically, authenticated BFD control packets in Poll Sequence.

### 3.5.1. Lightweight Authentication Mode Negotiation

Figure 7 displays the format of the Authentication field that is part of the Capability Encoding:

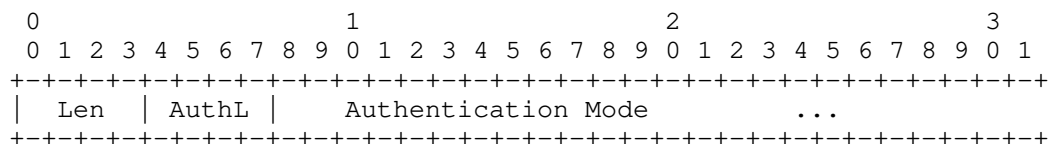


Figure 7: Lightweight Authentication Capability Field Format

where fields are defined as the following:

- o Len (Length) - four-bits long field. The value of the Length field is equal to the length of the Authentication field, including the Length, in octets.
- o AuthL (Authentication Length) - four bits size field. The value of the field is, in four octets long words, the longest

authentication signature the BFD system is capable of supporting for any of the methods advertised in the AuthMode field.

- o Authentication Mode - variable-length field. It is a bit-coded field that a BFD system uses to list modes of lightweight authentication it supports.

A BFD system uses Capability TLV, defined in Section 3.1, to discover the commonly supported mode of the Lightweight Authentication. The system sets the values in the Authentication field according to properly reflect its authentication capabilities. The BFD system transmits the Extended BFD control packet with Capability TLV as the first in a Poll Sequence. The remote BFD system that supports this specification receives the Extended BFD control packet with the advertised Lightweight Authentication modes and stores information locally. The system responds with the advertisement of its Lightweight Authentication capabilities in the Extended BFD control packet with the Final flag set. Each BFD system uses local and received information about Lightweight Authentication capabilities to deduce the commonly supported modes and selects from that set the one that uses the strongest authentication with the longest signature. If the common set is empty, i.e., none of supported by one BFD system authentication method is supported by another, an implementation **MUST** reflect this in its operational state and **SHOULD** notify an operator.

### 3.5.2. Using Lightweight Authentication

After BFD peers select an authentication mode for using in Lightweight Authentication each BFD system **MUST** use it to authenticate each Extended BFD control packet transmitted as part of a Poll Sequence using Lightweight Authentication TLV presented in Figure 8.

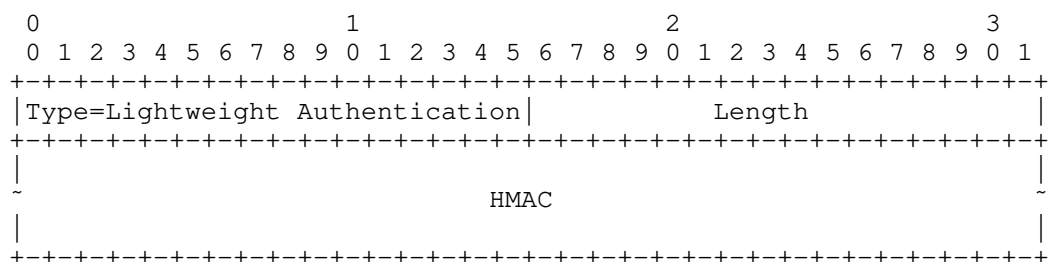


Figure 8: Lightweight Authentication TLV Format

where fields are defined as the following:

- o Type - TBA8 allocated by IANA in Section 4

- o Length - two octets long field equals length on the HMAC (Hashed Message Authentication Code) field in octets. The value of the Length field MUST be a multiple of 4.
- o HMAC - the hash value calculated on the entire preceding Extended BFD control packet data.

The Lightweight Authentication TLV MUST be the last TLV in an Extended BFD control packet. Padding TLV (Section 3.2) MAY be used to align the length of the Extended BFD control packet, excluding the Lightweight Authentication TLV, at multiple of 16 boundary.

The BFD system that receives the Extended BFD control packet with the Lightweight Authentication TLV MUST first validate the authentication by calculating the hash over the Extended BFD control packet. If the validation succeeds, the receiver MUST transmit the Extended BFD control packet with the Final flag set and the value of the Return code field in Diagnostic TLV set to None value (Table 5). If the validation of the lightweight authentication fails, then the BFD system MUST transmit the Extended BFD control packet with the Final flag set and the value of the Return Code field of the Diagnostic TLV set to Lightweight Authentication failed value (Table 5). The BFD system MUST have a control policy that defines actions when the system receives the Lightweight Authentication failed return code.

#### 4. IANA Considerations

##### 4.1. Extended BFD Message Types

IANA is requested to create the Extended BFD Message Types registry. All code points in the range 1 through 32759 in this registry shall be allocated according to the "IETF Review" procedure as specified in [RFC8126]. Code points in the range 32760 through 65279 in this registry shall be allocated according to the "First Come First Served" procedure as specified in [RFC8126]. Remaining code points are allocated according to Table 1:

| Value         | Description                  | Reference               |
|---------------|------------------------------|-------------------------|
| 0             | Reserved                     | This document           |
| 1- 32767      | Mandatory TLV,<br>unassigned | IETF Review             |
| 32768 - 65279 | Optional TLV,<br>unassigned  | First Come First Served |
| 65280 - 65519 | Experimental                 | This document           |
| 65520 - 65534 | Private Use                  | This document           |
| 65535         | Reserved                     | This document           |

Table 1: Extended BFD Type Registry

This document defines the following new values in Extended BFD Type registry:

| Value | Description                     | Reference     |
|-------|---------------------------------|---------------|
| TBA1  | Padding                         | This document |
| TBA2  | Capability                      | This document |
| TBA3  | Loss Measurement                | This document |
| TBA4  | Delay Measurement               | This document |
| TBA5  | Combined Loss/Delay Measurement | This document |
| TBA6  | Diagnostic                      | This document |
| TBA8  | Lightweight Authentication      | This document |

Table 2: Extended BFD Types

#### 4.2. Lightweight Authentication Modes

IANA is requested to create a Lightweight Authentication Modes registry. All code points in this registry shall be allocated according to the "IETF Review" procedure as specified in [RFC8126].

This document defines the following new values in the Lightweight Authentication Modes registry:



| Bit Position | Value | Description            | Reference     |
|--------------|-------|------------------------|---------------|
| 0            | 0x1   | Keyed SHA-1            | This document |
| 1            | 0x2   | Meticulous Keyed SHA-1 | This document |
| 2            | 0x4   | SHA-256                | This document |

Table 3: Lightweight Authentication Modes

#### 4.3. Return Codes

IANA is requested to create the Extended BFD Return Codes registry. All code points in the range 1 through 250 in this registry shall be allocated according to the "IETF Review" procedure as specified in [RFC8126]. Remaining code points are allocated according to Table 4:

| Value   | Description  | Reference     |
|---------|--------------|---------------|
| 0       | Reserved     | This document |
| 1- 250  | Unassigned   | IETF Review   |
| 251-253 | Experimental | This document |
| 254     | Private Use  | This document |
| 255     | Reserved     | This document |

Table 4: Extended BFD Return Codes Registry

This document defines the following new values in Extended BFD Return Codes registry:

| Value | Description                         | Reference     |
|-------|-------------------------------------|---------------|
| 0     | None                                | This document |
| 1     | One or more TLVs was not understood | This document |
| 2     | Lightweight Authentication failed   | This document |

Table 5: Extended BFD Return Codes

#### 5. Security Considerations

This document does not introduce new security aspects but inherits all security considerations from [RFC5880], [RFC6428], and [RFC6374].

## 6. References

### 6.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD)", RFC 5880, DOI 10.17487/RFC5880, June 2010, <<https://www.rfc-editor.org/info/rfc5880>>.
- [RFC6374] Frost, D. and S. Bryant, "Packet Loss and Delay Measurement for MPLS Networks", RFC 6374, DOI 10.17487/RFC6374, September 2011, <<https://www.rfc-editor.org/info/rfc6374>>.
- [RFC6428] Allan, D., Ed., Swallow, G., Ed., and J. Drake, Ed., "Proactive Connectivity Verification, Continuity Check, and Remote Defect Indication for the MPLS Transport Profile", RFC 6428, DOI 10.17487/RFC6428, November 2011, <<https://www.rfc-editor.org/info/rfc6428>>.
- [RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <<https://www.rfc-editor.org/info/rfc8126>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

### 6.2. Informative References

- [RFC8655] Finn, N., Thubert, P., Varga, B., and J. Farkas, "Deterministic Networking Architecture", RFC 8655, DOI 10.17487/RFC8655, October 2019, <<https://www.rfc-editor.org/info/rfc8655>>.

## Appendix A. Acknowledgements

TBD

Authors' Addresses

Greg Mirsky  
ZTE Corp.

Email: gregimirsky@gmail.com

Xiao Min  
ZTE Corp.

Email: xiao.min2@zte.com.cn

Network Working Group  
Internet-Draft  
Intended status: Informational  
Expires: July 15, 2021

H. Tian  
F. Zhao  
CAICT  
C. Xie  
China Telecom  
T. Li  
J. Ma  
China Unicom  
R. Mwehaire  
MTN Uganda Ltd.  
E. Chingwena  
MTN Group Limited  
S. Peng, Ed.  
Z. Li  
Y. Xiao  
Huawei Technologies  
January 11, 2021

SRv6 Deployment Consideration  
draft-tian-spring-srv6-deployment-consideration-04

Abstract

SRv6 has significant advantages over SR-MPLS and has attracted more and more attention and interest from network operators and verticals. Smooth network migration towards SRv6 is a key focal point and this document provides network design and migration guidance and recommendations on solutions in various scenarios. Deployment cases with SRv6 are also introduced.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

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."

This Internet-Draft will expire on July 15, 2021.

#### Copyright Notice

Copyright (c) 2021 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 (<https://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.

#### Table of Contents

|  |    |
|--|----|
| 1. Introduction . . . . .  | 3  |
| 2. Advantages of SRv6 . . . . .                                    | 4  |
| 2.1. IP Route Aggregation . . . . .                                | 4  |
| 2.2. End-to-end Service Auto-start . . . . .                       | 5  |
| 2.3. On-Demand Upgrade . . . . .                                   | 6  |
| 2.4. Simplified Service Deployment . . . . .                       | 7  |
| 2.4.1. Carrier's Carrier . . . . .                                 | 7  |
| 2.4.2. LDP over TE . . . . .                                       | 8  |
| 3. Compatibility Challenges . . . . .                              | 9  |
| 3.1. Fast Reroute (FRR) . . . . .                                  | 9  |
| 3.2. Traffic Engineering (TE) . . . . .                            | 10 |
| 3.3. Service Function Chaining (SFC) . . . . .                     | 10 |
| 3.4. IOAM . . . . .  | 10 |
| 4. Solutions for mitigating the compatibility challenges . . . . . | 11 |
| 4.1. Traffic Engineering . . . . .                                 | 12 |
| 4.1.1. Binding SID (BSID) . . . . .                                | 12 |
| 4.1.2. PCEP FlowSpec . . . . .                                     | 12 |
| 4.2. SFC . . . . .   | 12 |
| 4.2.1. Stateless SFC . . . . .                                     | 12 |
| 4.2.2. Stateful SFC . . . . .                                      | 13 |
| 4.3. Light Weight IOAM . . . . .                                   | 13 |
| 4.4. Postcard Telemetry . . . . .                                  | 14 |
| 5. Design Guidance for SRv6 Network . . . . .                      | 14 |
| 5.1. Locator and Address Planning . . . . .                        | 14 |

|  |    |
|--|----|
| 5.2. PSP . . . . .   | 15 |
| 6. Incremental Deployment Guidance for SRv6 Migration . . . . .  | 15 |
| 7. Migration Guidance for SRv6/SR-MPLS Co-existence Scenario . . | 16 |
| 8. Deployment cases . . . . .                                    | 17 |
| 8.1. China Telecom Si'chuan . . . . .                            | 18 |
| 8.2. China Unicom . . . . .                                      | 19 |
| 8.3. MTN Uganda . . . . .  | 20 |
| 9. IANA Considerations . . . . .                                 | 21 |
| 10. Security Considerations . . . . .                            | 21 |
| 11. Acknowledgement . . . . .                                    | 21 |
| 12. Contributors . . . . .                                       | 21 |
| 13. References . . . . .   | 22 |
| 13.1. Normative References . . . . .                             | 22 |
| 13.2. Informative References . . . . .                           | 22 |
| Authors' Addresses . . . . .                                     | 24 |

## 1. Introduction

SRv6 is the instantiation of Segment Routing deployed on the IPv6 data plane [RFC8200]. Therefore, in order to support SRv6, the network must first be enabled for IPv6. Over the past several years, IPv6 has been actively promoted all over the world, and the deployments of IPv6 have been ever-increasing which provides the basis for the deployments of SRv6.

With IPv6 as its data plane, for network migration towards SRv6, both software and hardware need to be upgraded. Compared with other new protocols, only IGP and BGP need to be extended to support SRv6, which significantly simplifies the software upgrade required. While the hardware needs to support the new SRv6 header SRH [RFC8754], the design of SRv6 assures compatibility with the existing IPv6 network as an SRv6 SID is designed as a 128-bit IPv6 address and the encapsulation of an SRv6 packet is the same as an IPv6 packet. When only L3VPN over SRv6 BE (Best-Effort) is deployed, there will be no SRH. Therefore, no additional hardware capabilities are required but only software upgrade for protocol extensions.

As the number of services supported by SRv6 increase, e.g. SFC, network slicing, iOAM etc., more SIDs in the SRH may impose new requirements on the hardware. Besides upgrading the hardware, various solutions have already been proposed to relieve the imposed pressure on the hardware, such as Binding SID (BSID) etc. to guarantee the compatibility with the existing network. On the other hand SRv6 has many more advantages over SR-MPLS for the network migration to support new services.

This document summarizes the advantages of SRv6 and provides network migration guidance and recommendations on solutions in various scenarios.

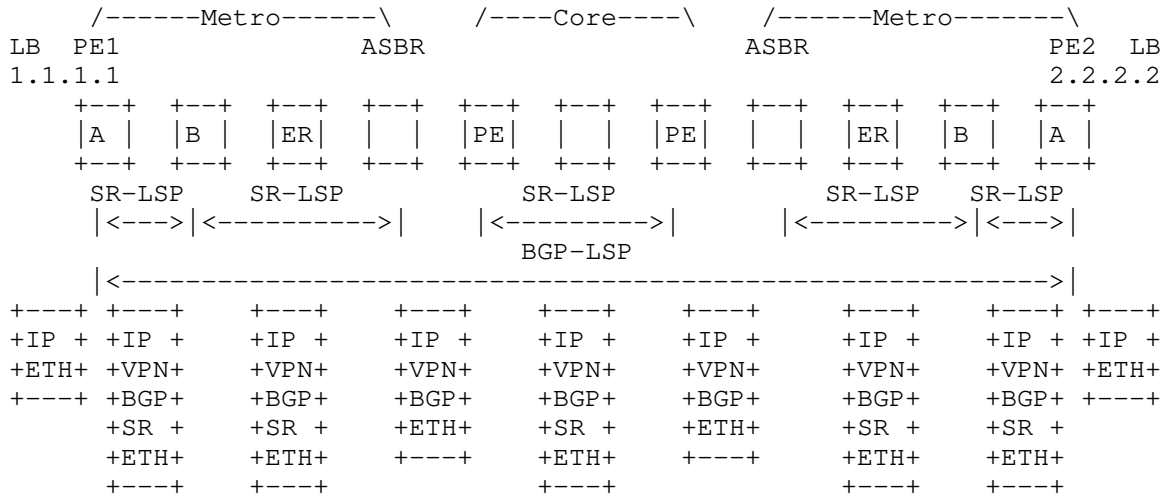
## 2. Advantages of SRv6

Compared with SR-MPLS, SRv6 has significant advantages especially in large scale networking scenarios.

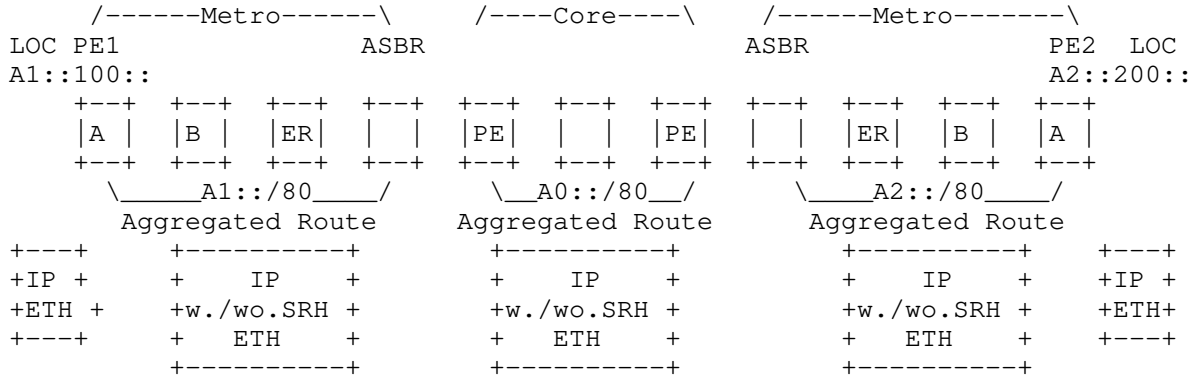
### 2.1. IP Route Aggregation

The increasing complexity of service deployment is of concern for network operators, especially in large-scale networking scenarios. With solutions such as multi-segment PW and Option A [RFC4364], the number of service-touch points has increased, and the services, with associated OAM features cannot be deployed end-to-end.

- o With Seamless MPLS or SR-MPLS, since the MPLS label itself does not have reachability information, it must be attached to a routable address. The 32-bit host route needs to leak across domains. For an extreme case, as shown in Figure 1a, in a large scale networking scenario, millions of host route LSPs might need to be imported, which places big challenges on the capabilities of the edge nodes.
- o With SRv6, owing to its native IP feature of route aggregation as shown in Figure 1b, the aggregated routes can be imported across network domains. For large scale networking, only very few aggregated routes are needed in order to start end-to-end services, which also reduces the scalability requirements on the edge nodes.



(a) SR-MPLS



(b) SRv6

Figure 1. Large-scale Networking with (a) SR-MPLS vs. (b) SRv6

## 2.2. End-to-end Service Auto-start

In the SR cross-domain scenario, in order to set up end-to-end SR tunnels, the SIDs in each domain need to be imported to other domains.

- o With SR-MPLS, SRGB and Node SID need overall network-wide planning, and in the cross-domain scenario, it is difficult or sometimes even impossible to perform as the node SIDs in different



domains may collide. BGP Prefix SID can be used for the cross-domain SID import, but the network operator must be careful when converting the SID to avoid SID collision. Moreover, the pre-allocated SRGB within each domain needs to consider the total number of devices in all other domains, which raises difficulties for the network-wide planning.

- o With SRv6, owing to its native IP feature of route reachability, if the IPv6 address space is carefully planned, and the aggregated routes are imported by using BGP4+ (BGP IPv6), the services will auto-start in the cross-domain scenario.

### 2.3. On-Demand Upgrade

The MPLS label itself does not hold any reachability information, so it must be attached to a routable address, which means that the matching relationship between the label and FEC needs to be maintained along the path.

SR-MPLS uses the MPLS data plane. When the network migrates to SR-MPLS, there are two ways, as shown in Figure 2:

1. MPLS/SR-MPLS Dual stack: the entire network is upgraded first and then deploy SR-MPLS.
2. MPLS and SR-MPLS interworking: mapping servers are deployed at some of the intermediate nodes and then removed once the entire network is upgraded

Regardless of which migration option is chosen, big changes in a wide area is required at the initial stage therefore causing a long time-to-market.

In contrast, the network can be migrated to SRv6 on demand. Wherever the services need to be turned on, only the relevant devices need to be upgraded to enable SRv6, and all other devices only need to support IPv6 forwarding and need not be aware of SRv6. When Traffic Engineering (TE) services are needed, only the key nodes along the path need to be upgraded to support SRv6.

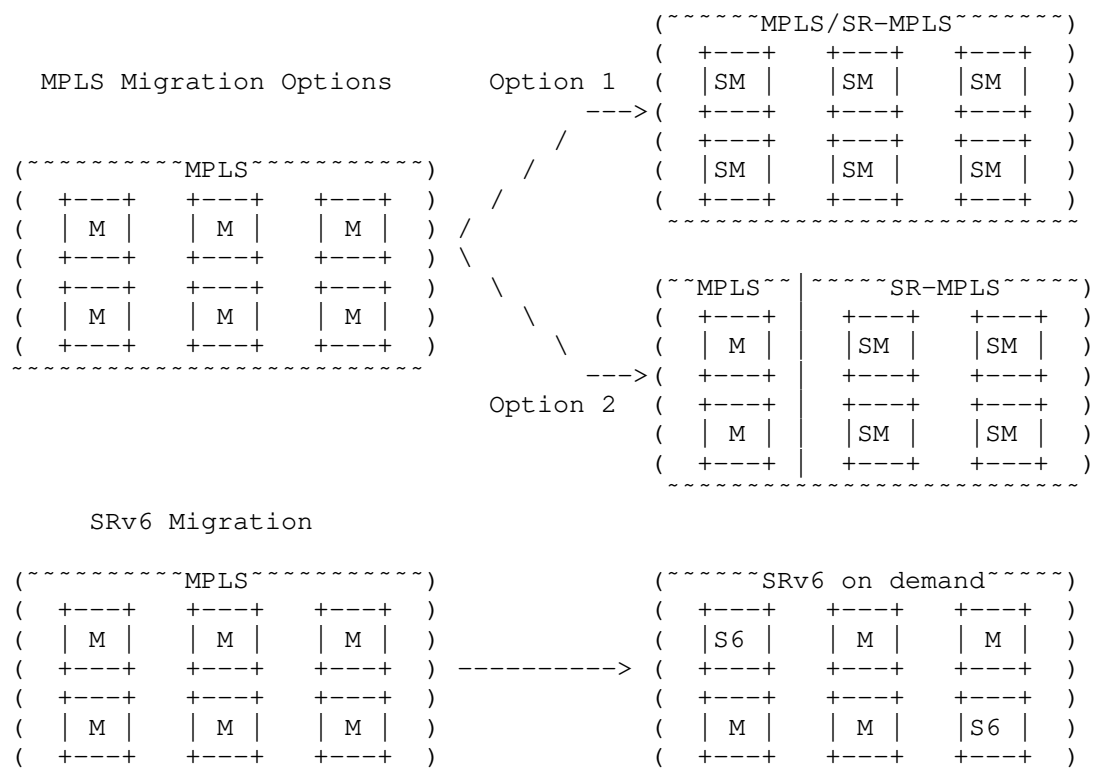


Figure 2. MPLS Domain Migration vs. SRv6 On-Demand Upgrade

#### 2.4. Simplified Service Deployment

With SRv6, the service deployment can be significantly simplified in some scenarios.

##### 2.4.1. Carrier's Carrier

When the customer of the VPN service carrier (Provider Carrier) is itself a VPN service carrier (Customer Carrier), it becomes the scenario of Carrier's Carrier. For this scenario, with SRv6, the service deployment can be significantly simplified.

To achieve better scalability, the CEs of the Provider Carrier (i.e. the PEs of the Customer Carriers) only distribute the internal network routes to the PEs of the Provider Carrier. The customers' routes of the Customer Carriers (i.e. from CE3 and CE4) will not be distributed into the network of the Provide Carrier. Therefore, LDP or Labeled BGP will be run between the CEs of the Provider Carrier

(i.e. CE1 and CE2 in the Figure 3) and the PEs of the Provider Carrier (i.e. PE1 and PE2 in the Figure 3), and LDP will be run between the CEs of the Provider Carrier (i.e. the PEs of the Customer Carriers) and the PEs of the Customer Carrier (i.e. PE3 and PE4 in the Figure 3). MP-BGP will be run between the PEs of the Customer Carrier. The overall service deployment is very complex.

If SRv6 is deployed by the Customer Carrier and the Provider Carrier, no LDP will be ever needed. The Locator routes and Loopback routes of the Customer Carriers can be distributed into the network of the Provider Carrier via BGP, and within each carrier's network only IGP is needed. The end-to-end VPN services can be provided just based on the IPv6 interconnections, and the customer carrier is just like a normal CE to the provider carrier, which significantly simplified the VPN service deployment.

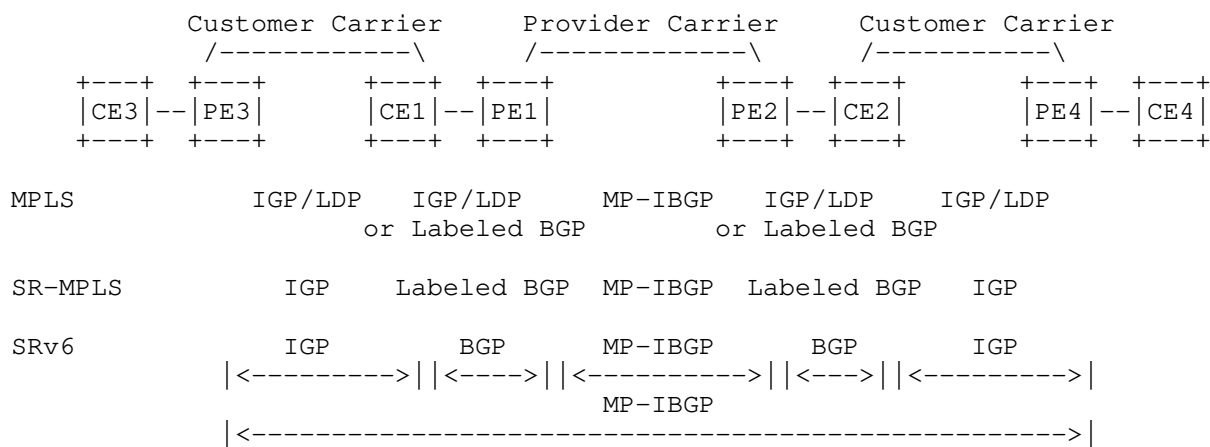


Figure 3. Service deployment with MPLS, SR-MPLS and SRv6

#### 2.4.2. LDP over TE

In a MPLS network, generally RSVP-TE is deployed in the P nodes of the network, and LDP is running between these P nodes and the PE nodes. Customers access to VPN services via the PE nodes. This scenario is called LDP over TE, which is a typical deployment for carriers who want to achieve the TE capability over MPLS network while keep scalability. However, such network configuration and service deployment are very complex.

With SRv6 which can provide both TE capability and IP reachability, the service deployment can be significantly simplified. Only IGP and BGP are needed in the network to launch VPN services.

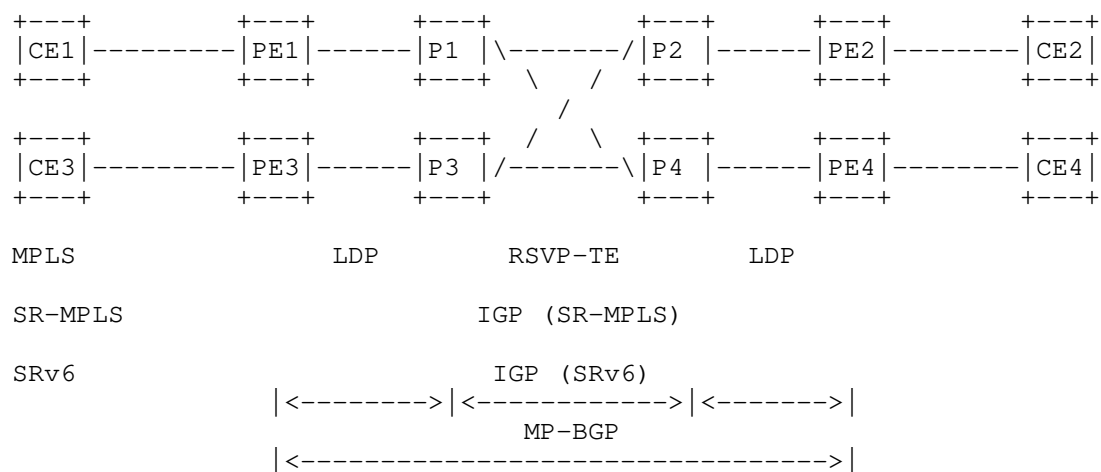


Figure 4. Service deployment with (a) MPLS/SR-MPLS vs. (b) SRv6

### 3. Compatibility Challenges

By adopting SR Policy, state in the network devices can be greatly reduced, which ultimately evolves the network into a stateless fabric. However, it also brings compatibility challenges on the legacy devices. In particular, the legacy devices need to upgrade software and/or hardware in order to support the processing of SRH.

Furthermore, as the segments in the segment list increase the SR Policy incrementally expands, the encapsulation header overhead increases, which imposes high performance requirements on the performance of hardware forwarding (i.e. the capability of the chipset).

This section identifies the challenges for legacy devices imposed by SRv6 in the following SPRING use cases.

#### 3.1. Fast Reroute (FRR)

FRR is deployed to cope with link or node failures by precomputing backup paths. By relying on SR, Topology Independent Loop-free Alternate Fast Re-route (TI-LFA)

[I-D.ietf-rtgwg-segment-routing-ti-lfa] provides a local repair mechanism with the ability to activate the data plane switch-over on to a loop-free backup path irrespective of topologies prior and after the failure.

Using SR, there is no need to create state in the network in order to enforce FRR behavior. Correspondingly, the Point of Local Repair,

i.e. the protecting router, needs to insert a repair list at the head of the segment list in the SRH, encoding the explicit post-convergence path to the destination. This action will increase the length of the segment list in the SRH as shown in Figure 1.

### 3.2. Traffic Engineering (TE)

TE enables network operators to control specific traffic flows going through configured explicit paths. There are loose and strict options. With the loose option, only a small number of hops along the path is explicitly expressed, while the strict option specifies each individual hop in the explicit path, e.g. to encode a low latency path from one network node to another.

With SRv6, the strict source-routed explicit paths will result in a long segment list in the SRH as shown in Figure 1, which places high requirements on the devices.

### 3.3. Service Function Chaining (SFC)

The SR segments can also encode instructions, called service segments, for steering packets through services running on physical service appliances or virtual network functions (VNF) running in a virtual environment [I-D.ietf-spring-sr-service-programming]. These service segments can also be integrated in an SR policy along with node and adjacency segments. This feature of SR will further increase the length of the segment list in the SRH as shown in Figure 1.

In terms of SR awareness, there are two types of services, i.e. SR-aware and SR-unaware services, which both impose new requirements on the hardware. The SR-aware service needs to be fully capable of processing SR traffic, while for the SR-unaware services, an SR proxy function needs to be defined.

If the Network Service Header (NSH) based SFC [RFC8300] has already been deployed in the network, the compatibility with existing NSH is required.

### 3.4. IOAM

IOAM, i.e. "in-situ" Operations, Administration, and Maintenance (OAM), encodes telemetry and operational information within the data packets to complement other "out-of-band" OAM mechanisms, e.g. ICMP and active probing. The IOAM data fields, i.e. a node data list, hold the information collected as the packets traverse the IOAM domain [I-D.ietf-ippm-ioam-data], which is populated iteratively starting with the last entry of the list.

The IOAM data can be embedded into a variety of transports. To support the IOAM on the SRv6 data plane, the O-flag in the SRH is defined [I-D.ietf-6man-spring-srv6-oam], which implements the "punt a timestamped copy and forward" or "forward and punt a timestamped copy" behavior. The IOAM data fields, i.e. the node data list, are encapsulated in the IOAM TLV in SRH, which further increases the length of the SRH as shown in Figure 1.

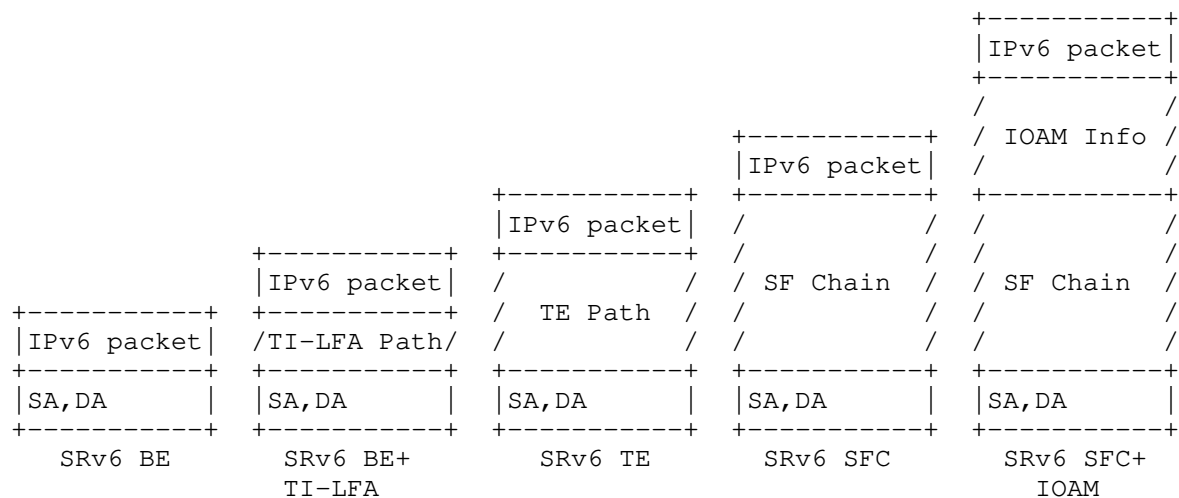


Figure 1. Evolution of SRv6 SRH

Compatibility challenges for legacy devices can be summarized as follows:

- o Legacy devices need to upgrade software and/or hardware in order to support the processing of SRH
- o As the SRH expands, the encapsulation overhead increases and correspondingly the effective payload decreases
- o As the SRH expands, the hardware forwarding performance reduces which requires higher capabilities of the chipset

#### 4. Solutions for mitigating the compatibility challenges

This section provides solutions to mitigate the challenges outlined in section 2.

#### 4.1. Traffic Engineering

With strict traffic engineering, the resultant long SID list in the SRH raises high requirements on the hardware chipset, which can be mitigated by the following solutions.

##### 4.1.1. Binding SID (BSID)

Binding SID [RFC8402] involves a list of SIDs and is bound to an SR Policy. The node(s) that imposes the bound policy needs to store the SID list. When a node receives a packet with its active segment as a BSID, the node will steer the packet in to the bound policy accordingly.

To reduce the long SID list of a strict TE explicit path, BSID can be used at selective nodes, maybe according to the processing capacity of the hardware chipset. BSID can also be used to impose the repair list in the TI-LFA as described in Section 2.1.

##### 4.1.2. PCEP FlowSpec

When the SR architecture adopts a centralized model, the SDN controller (e.g. Path Computation Element (PCE)) only needs to apply the SR policy at the head-end. There is no state maintained at midpoints and tail-ends. Eliminating state in the network (midpoints and tail-points) is a key benefit of utilizing SR. However, it also leads to a long SID list for expressing a strict TE path.

PCEP FlowSpec [I-D.ietf-pce-pcep-flowspec] provides a trade-off solution. PCEP FlowSpec is able to disseminate Flow Specifications (i.e. filters and actions) to indicate how the classified traffic flows will be treated. In an SR-enabled network, PCEP FlowSpec can be applied at the midpoints to enforce traffic engineering policies where it is needed. In that case, state needs to be maintained at the corresponding midpoints of a TE explicit path, but the SID list can be shortened.

#### 4.2. SFC

Currently two approaches are proposed to support SFC over SRv6, i.e. stateless SFC [I-D.ietf-spring-sr-service-programming] and stateful SFC [I-D.ietf-spring-nsh-sr].

##### 4.2.1. Stateless SFC

A service can also be assigned an SRv6 SID which is integrated into an SR policy and used to steer traffic to it. In terms of the capability of processing the SR information in the received packets,

there are two types of services, i.e. SR-aware service and SR-unaware service. An SR-aware service can process the SRH in the received packets. An SR-unaware service, i.e. legacy service, is not able to process the SR information in the traffic it receives, and may drop the received packets. In order to support such services in an SRv6 domain, the SR proxy is introduced to handle the processing of SRH on behalf of the SR-unaware service. The service SID associated with the SR-unaware service is instantiated on the SR proxy, which is used to steer traffic to the service.

The SR proxy intercepts the SR traffic destined for the service via the locally instantiated service SID, removes the SR information, and sends the non-SR traffic out on a given interface to the service. When receiving the traffic coming back from the service, the SR proxy will restore the SR information and forwards it to the next segment in the segment list.

#### 4.2.2. Stateful SFC

The NSH and SR can be integrated in order to support SFC in an efficient and cost-effective manner while maintaining separation of the service and transport planes.

In this NSH-SR integration solution, NSH and SR work jointly and complement each other. Specifically, SR is responsible for steering packets along a given Service Function Path (SFP) while NSH is for maintaining the SFC instance context, i.e. Service Path Identifier (SPI), Service Index (SI), and any associated metadata.

When a service chain is established, a packet associated with that chain will be first encapsulated with an NSH and then an SRH, and forwarded in the SR domain. When the packet arrives at an SFF and needs to be forwarded to an SF, the SFF performs a lookup based on the service SID associated with the SF to retrieve the next-hop context (a MAC address) between the SFF and SF. Then the SFF strips the SRH and forwards the packet with NSH carrying metadata to the SF where the packet will be processed as specified in [RFC8300]. In this case, the SF is not required to be capable of the SR operation, neither is the SR proxy. Meanwhile, the stripped SRH will be updated and stored in a cache in the SFF, indexed by the NSH SPI for the forwarding of the packet coming back from the SF.

#### 4.3. Light Weight IOAM

In most cases, after the IPv6 Destination Address (DA) is updated according to the active segment in the SRH, the SID in the SRH will not be used again. However, the entire SID list in the SRH will



still be carried in the packet along the path till a PSP/USP is enforced.

The light weight IOAM method [I-D.li-spring-passive-pm-for-srv6-np] makes use of the used segments in the SRH to carry the IOAM information, which saves the extra space in the SRH and mitigate the requirements on the hardware.

#### 4.4. Postcard Telemetry

Existing in-situ OAM techniques incur encapsulation and header overhead issues as described in section 2. Postcard-based Telemetry with Packet Marking for SRv6 on-path OAM[I-D.song-ippm-postcard-based-telemetry], provides a solution that avoids the extra overhead for encapsulating telemetry-related instruction and metadata in SRv6 packets.

### 5. Design Guidance for SRv6 Network

#### 5.1. Locator and Address Planning

Address Planning is a very important factor for a successful network design, especially an IPv6 network, which will directly affect the design of routing, tunnel, and security. A good address plan can bring big benefit for service deployment and network operation.

If a network has already deployed IPv6 and set up IPv6 subnets, one of the subnets can be selected for the SRv6 Locator planning, and the existing IPv6 address plan will not be impacted.

If a network has not yet deployed IPv6 and there has not been an address plan, it needs to perform the IPv6 address planning first taking the following steps,

1. to decide the IPv6 address planning principles
2. to choose the IPv6 address assignment methods
3. to assign the IPv6 address in a hierarchical manner

For an SRv6 network, in the first step for IPv6 address planning, the following principles are suggested to follow,

1. Unification: all the IPv6 addresses SHOULD be planned altogether, including service addresses for end users, platform addresses (for IPTV, DHCP servers), and network addresses for network devices interconnection.

2. Uniqueness: every single address SHOULD be unique.
3. Separation: service addresses and network addresses SHOULD be planned separately; the SRv6 Locator subnet, the Loopback interface addresses and the link addresses SHOULD be planned separately.
4. Aggregatability: when being distributed across IGP/BGP domains, the addresses in the preassigned subnets (e.g. SRv6 Locator subnet, the Loopback interface subnet) SHOULD be aggregatable, which will make the routing easier.
5. Security: fast tracability of the assigned addresses SHOULD be facilitated, which will make the traffic filtering easier.
6. Evolvability: enough address space SHOULD be reserved for each subset for future service development.

Considering the above-mentioned IPv6 address planning principles, it has been adopted in some deployment cases to set Locator length 96bits, function length 20bits, and args 12bits.

## 5.2. PSP

When Locator is imported in ISIS, the system will automatically assign END SID with Flavors such as PSP (Penultimate Segment Pop) and distribute the Locator subnet route through ISIS.

The Flavor PSP, that is, SRH is popped at penultimate segment, provides the following benefits,

1. Reduce the load of ultimate segment endpoint. Ultimate segment endpoint tends to have heavy load since it needs to handle the inner IP/IPv6/Ethernet payload and demultiplex the packet to the right overlay service.
2. Support of incremental deployment on existing network where the ultimate segment endpoint is low-end device that is not fully capable of handling SRH.

## 6. Incremental Deployment Guidance for SRv6 Migration

Incremental deployment is the key for a smooth network migration to SRv6. In order to quickly launch SRv6 network services and enjoy the benefits brought by SRv6, the recommended incremental SRv6 deployment steps are given as follows. These are based on practical deployment experience earned from the use cases described in [I-D.matsushima-spring-srv6-deployment-status].

The referenced network topology is shown in Figure 5.

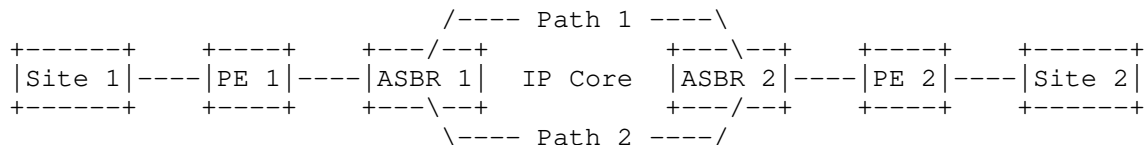


Figure 5. Reference Network Topology

Step1. All the network devices are upgraded to support IPv6.

Step 2. According to service demands, only a set of selected PE devices are upgraded to support SRv6 in order to immediately deploy SRv6 overlay VPN services. For instance, in Figure 3, PE1 and PE2 are SRv6-enabled.

Step 3. Besides the PE devices, some P devices are upgraded to support SRv6 in order to deploy loose TE which enables network path adjustment and optimization. SFC is also a possible service provided by upgrading some of the network devices.

Step 4. All the network devices are upgraded to support SRv6. In this case, it is now possible to deploy strict TE, which enables the deterministic networking and other strict security inspection.

## 7. Migration Guidance for SRv6/SR-MPLS Co-existence Scenario

As the network migration to SRv6 is progressing, in most cases SRv6-based services and SR-MPLS-based services will coexist.

As shown in Figure 6, in the Non-Standalone (NSA) case specified by 3GPP Release 15, 5G networks will be supported by existing 4G infrastructure. 4G eNB connects to CSG 2, 5G gNB connects to CSG 1, and EPC connects to RSG 1.

To support the 4G services, network services need to be provided between CSG 2 and RSG 1 for interconnecting 4G eNB and EPC, while for the 5G services, network services need to be deployed between CSG 1 and RSG 1 for interconnecting 5G gNB and EPC. Meanwhile, to support X2 interface between the eNB and gNB, network services also need to be deployed between the CSG 1 and CSG 2.

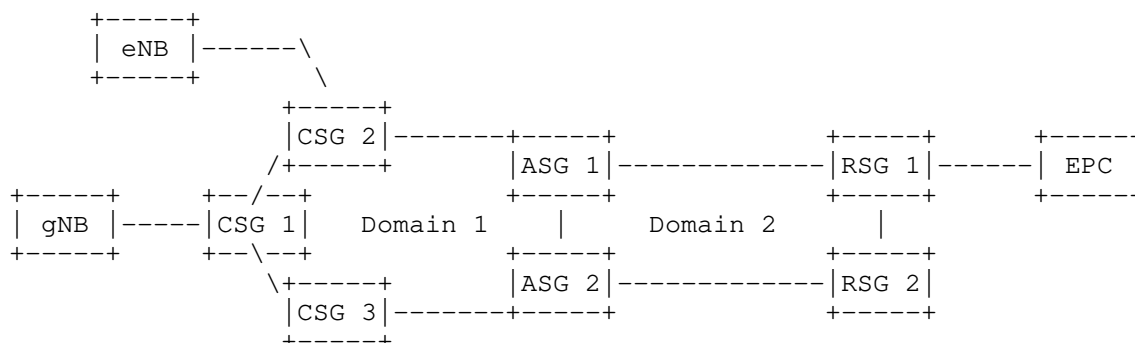


Figure 6. A 3GPP Non-Standalone deployment case

As shown in Figure 6, in most of the current network deployments, MPLS-based network services may have already existed between CSG 2 and RSG 1 for interconnecting 4G eNB and EPC for 4G services.

When 5G services are to be supported, more stringent network services are required, e.g. low latency and high bandwidth. SRv6-based network services could be deployed between CSG 1 and RSG 1 for interconnecting 5G gNB and EPC.

In order to perform smooth network migration, a dual-stack solution can be adopted which deploys both SRv6 and MPLS stack in one node.

With the dual-stack solution, only CSG 1 and RSG 1 need to be upgraded with SRv6/MPLS dual stack. In this case, CSG 1 can immediately start SRv6-based network services to RSG 1 for support of 5G services, but continue to use MPLS-based services to CSG 2 for X2 interface communications. The upgrade at CSG 1 will not affect the existing 4G services supported by the MPLS-based network services between CSG 2 and RSG 1. RSG1 can provide MPLS services to CSG2 for 4G services as well as SRv6 services to CSG 1 for 5G services.

## 8. Deployment cases

With the current network, the launch of leased line service is slow, the network operation and maintenance is complex, and the configuration points are many. SRv6 can solve the issues above. There have already been several successful SRv6 deployments following the incremental deployment guidance shown in Section 3.

### 8.1. China Telecom Si'chuan

China Telecom Si'chuan (Si'chuan Telecom) has enabled SRv6 at the PE node of the Magic-Mirror DC in Mei'shan, Cheng'du, Pan'zhihua and other cities. The SRv6 BE tunnel has been deployed through the 163 backbone network which has the IPv6 capability. It enables the fast launch of the Magic-Mirror video service, the interconnection of the DCs in various cities, and the isolation of video services. The deployment case is shown in Figure 7.

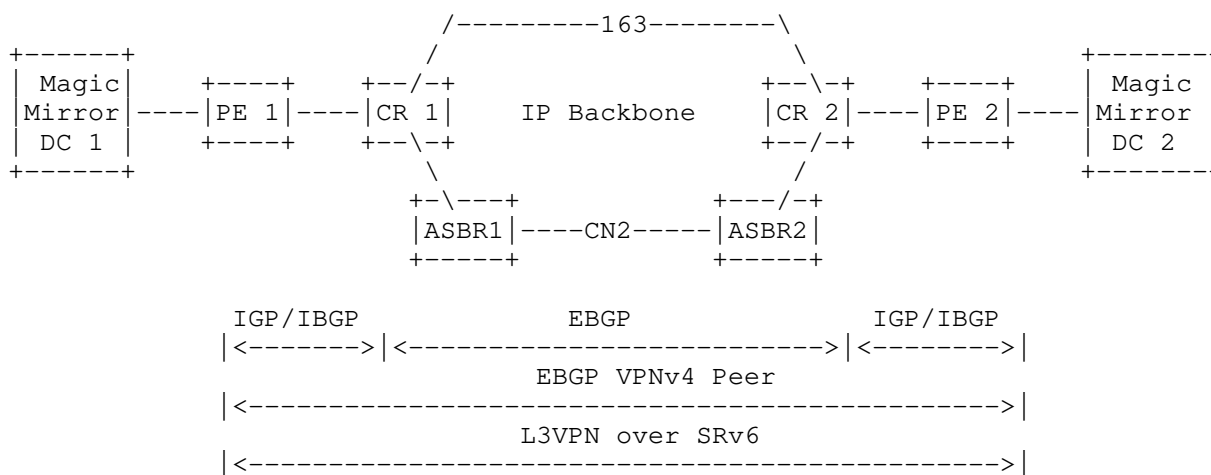


Figure 7. China Telecom Si'chuan deployment case

As shown in Figure 7, IGP (some cities such as Chengdu deploy ISIS, while other cities such as Panzhihua deploy OSPF) and IBGP are deployed between PE and CR, and EBGp is deployed between CRs of cities in order to advertise the aggregation route. EBGp VPNv4 peers are set up between PEs in different cities to deliver VPN private network routes.

The packet enters the SRv6 BE tunnel from the egress PE of DC, and the packet is forwarded according to the Native IP of the 163 backbone network. When the packet reaches the peer PE, the SRH is decapsulated, and then the IP packet is forwarded in the VRF according to the service SID (for example, End.DT4).

In order to further implement the path selection, ASBRs can be upgraded to support SRv6. Different SRv6 policies are configured on the DC egress PE so that different VPN traffic reaches the peer PE

through the 163 backbone network and the CN2 backbone network respectively.

## 8.2. China Unicom

China Unicom has deployed SRv6 L3VPN over 169 IPv6 backbone network from Guangzhou to Beijing to provide inter-domain Cloud VPN service. The deployment case is shown in Figure 8.

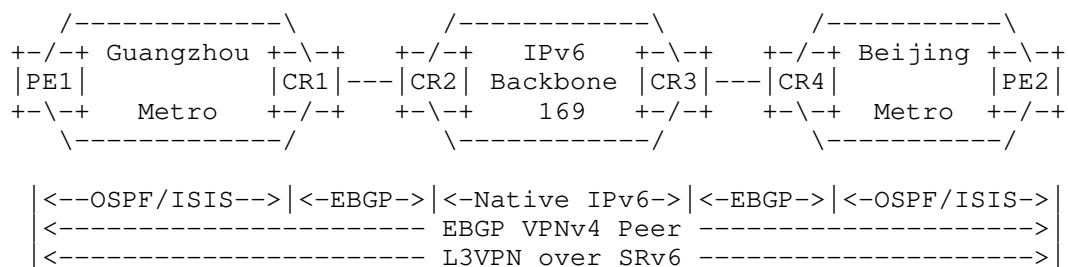


Figure 8. China Unicom SRv6 L3VPN case

In Guangzhou and Beijing metro networks, routers exchange basic routing information using IGP(OSPF/ISIS). The prefixes of IPv6 loopback address and SRv6 locator of routers are different, and both of them need to be imported into the IGP. The 169 backbone is a native IPv6 network. Between metro and backbone, the border routers establish EBGPeer with each other, e.g. CR1 with CR2, CR3 with CR4, to form basic connectivity. All of these constitute the foundation of overlay services, and have not been changed.

PE1 and PE2 establish EBGPeer and advertise VPNv4 routes with each other. If one site connects to two PEs, metro network will use multi RD, community and local preference rules to choose one best route and one backup.

After basic routing among networks and VPN routes between the two PEs are all ready, two PEs encapsulate and forward VPN traffic within SRv6 tunnel. The tunnel is SRv6 best effort (BE) tunnel. It introduces only outer IPv6 header but not SRH header into traffic packets. After encapsulation, the packet is treated as common IPv6 packet and forwarded to the egress PE, which performs decapsulation and forwards the VPN traffic according to specific VRF.

Guangdong Unicom has also launched the SRv6 L3VPN among Guangzhou, Shenzhen, and Dongguan, which has passed the interop test between different vendors.

With SRv6 enabled at the PE devices, the VPN service can be launched very quickly without impact on the existing traffic. With SRv6 TE further deployed, more benefits of using SRv6 can be exploited.

### 8.3. MTN Uganda

MTN Uganda has enabled SRv6 at the MPBN PE/P nodes. The SRv6 BE tunnel has been deployed through the MPBN network which has the IPv6 capability. It enables the fast service provisioning for mobile service, enterprise service and internal IT services, and also improves service SLA such as service monitoring and availability. The deployment case is shown in Figure 9.

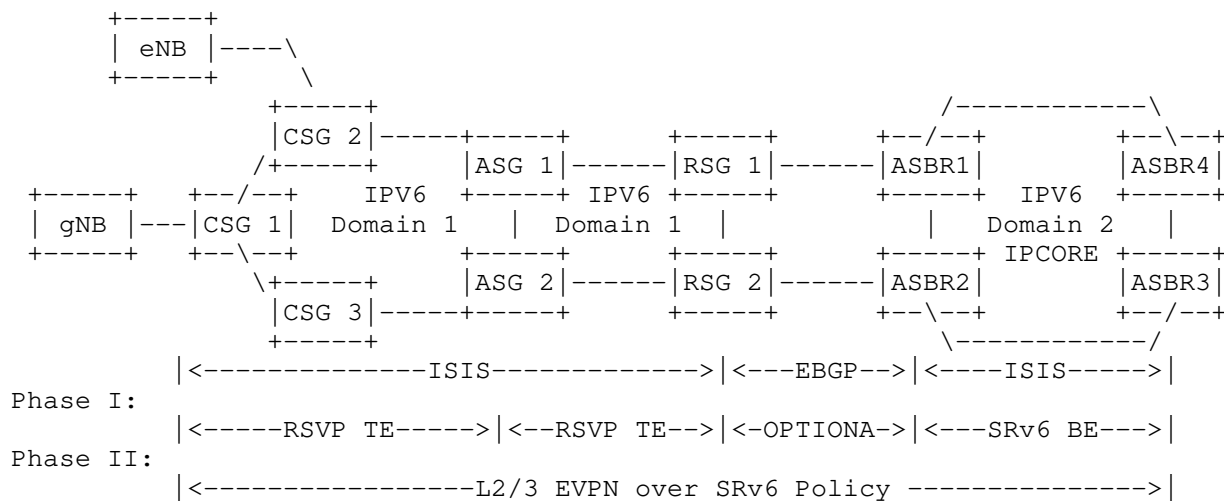


Figure 9. MTN Uganda Deployment Case

As shown in the Figure 9,

In the phase I, SRv6 BE was deployed in MPBN network. All services in the MPBN will be carried through SRv6 BE in the core network. The Option A is deployed between the IPRAN network and Core network.

In the phase II, SRv6 Policy will be deployed E2E from IPRAN to Core. Cross-domain path selection is available for mobile and enterprise services. The service will be carried in SRv6 Policy through the entire MPBN network.

L3VPN and L2VPN services will evolve to EVPN to simplify the network operation and management.

## 9. IANA Considerations

There are no IANA considerations in this document.

## 10. Security Considerations

TBD.

## 11. Acknowledgement

The section on the PSP use cases is inspired from the discussions over the mailing list. The authors would like to acknowledge the constructive discussions from Daniel Voyer, Jingrong Xie, etc..

## 12. Contributors

Hailong Bai  
China Unicom  
China

Email:

Jichun Ma  
China Unicom  
China

Email:

Huizhi Wen  
Huawei Technologies  
China

Email: wenhuizhi@huawei.com

Ruizhao Hu  
Huawei Technologies  
China

Email: huruizhao@huawei.com

Jianwei Mao  
Huawei  
China

Email: maojianwei@huawei.com



## 13. References

### 13.1. Normative References

- [I-D.ietf-spring-srv6-network-programming]  
Filsfils, C., Camarillo, P., Leddy, J., Voyer, D.,  
Matsushima, S., and Z. Li, "SRv6 Network Programming",  
draft-ietf-spring-srv6-network-programming-28 (work in  
progress), December 2020.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate  
Requirement Levels", BCP 14, RFC 2119,  
DOI 10.17487/RFC2119, March 1997,  
<<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private  
Networks (VPNs)", RFC 4364, DOI 10.17487/RFC4364, February  
2006, <<https://www.rfc-editor.org/info/rfc4364>>.
- [RFC5659] Bocci, M. and S. Bryant, "An Architecture for Multi-  
Segment Pseudowire Emulation Edge-to-Edge", RFC 5659,  
DOI 10.17487/RFC5659, October 2009,  
<<https://www.rfc-editor.org/info/rfc5659>>.
- [RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6  
(IPv6) Specification", STD 86, RFC 8200,  
DOI 10.17487/RFC8200, July 2017,  
<<https://www.rfc-editor.org/info/rfc8200>>.
- [RFC8754] Filsfils, C., Ed., Dukes, D., Ed., Previdi, S., Leddy, J.,  
Matsushima, S., and D. Voyer, "IPv6 Segment Routing Header  
(SRH)", RFC 8754, DOI 10.17487/RFC8754, March 2020,  
<<https://www.rfc-editor.org/info/rfc8754>>.

### 13.2. Informative References

- [I-D.ietf-6man-segment-routing-header]  
Filsfils, C., Dukes, D., Previdi, S., Leddy, J.,  
Matsushima, S., and D. Voyer, "IPv6 Segment Routing Header  
(SRH)", draft-ietf-6man-segment-routing-header-26 (work in  
progress), October 2019.
- [I-D.ietf-6man-spring-srv6-oam]  
Ali, Z., Filsfils, C., Matsushima, S., Voyer, D., and M.  
Chen, "Operations, Administration, and Maintenance (OAM)  
in Segment Routing Networks with IPv6 Data plane (SRv6)",  
draft-ietf-6man-spring-srv6-oam-08 (work in progress),  
October 2020.

- [I-D.ietf-ippm-ioam-data]  
Brockners, F., Bhandari, S., and T. Mizrahi, "Data Fields for In-situ OAM", draft-ietf-ippm-ioam-data-11 (work in progress), November 2020.
- [I-D.ietf-pce-pcep-flowspec]  
Dhody, D., Farrel, A., and Z. Li, "PCEP Extension for Flow Specification", draft-ietf-pce-pcep-flowspec-12 (work in progress), October 2020.
- [I-D.ietf-rtgwg-segment-routing-ti-lfa]  
Litkowski, S., Bashandy, A., Filsfils, C., Decraene, B., and D. Voyer, "Topology Independent Fast Reroute using Segment Routing", draft-ietf-rtgwg-segment-routing-ti-lfa-05 (work in progress), November 2020.
- [I-D.ietf-spring-nsh-sr]  
Guichard, J. and J. Tantsura, "Integration of Network Service Header (NSH) and Segment Routing for Service Function Chaining (SFC)", draft-ietf-spring-nsh-sr-04 (work in progress), December 2020.
- [I-D.ietf-spring-sr-service-programming]  
Clad, F., Xu, X., Filsfils, C., daniel.bernier@bell.ca, d., Li, C., Decraene, B., Ma, S., Yadlapalli, C., Henderickx, W., and S. Salsano, "Service Programming with Segment Routing", draft-ietf-spring-sr-service-programming-03 (work in progress), September 2020.
- [I-D.li-spring-passive-pm-for-srv6-np]  
Li, C. and M. Chen, "Passive Performance Measurement for SRv6 Network Programming", draft-li-spring-passive-pm-for-srv6-np-00 (work in progress), March 2018.
- [I-D.matsushima-spring-srv6-deployment-status]  
Matsushima, S., Filsfils, C., Ali, Z., Li, Z., and K. Rajaraman, "SRv6 Implementation and Deployment Status", draft-matsushima-spring-srv6-deployment-status-10 (work in progress), December 2020.
- [I-D.song-ippm-postcard-based-telemetry]  
Song, H., Zhou, T., Li, Z., Mirsky, G., Shin, J., and K. Lee, "Postcard-based On-Path Flow Data Telemetry using Packet Marking", draft-song-ippm-postcard-based-telemetry-08 (work in progress), October 2020.

- [RFC8300] Quinn, P., Ed., Elzur, U., Ed., and C. Pignataro, Ed.,  
"Network Service Header (NSH)", RFC 8300,  
DOI 10.17487/RFC8300, January 2018,  
<<https://www.rfc-editor.org/info/rfc8300>>.
- [RFC8402] Filsfils, C., Ed., Previdi, S., Ed., Ginsberg, L.,  
Decraene, B., Litkowski, S., and R. Shakir, "Segment  
Routing Architecture", RFC 8402, DOI 10.17487/RFC8402,  
July 2018, <<https://www.rfc-editor.org/info/rfc8402>>.

## Authors' Addresses

Hui Tian  
CAICT  
China

Email: [tianhui@caict.ac.cn](mailto:tianhui@caict.ac.cn)

Feng Zhao  
CAICT  
China

Email: [zhaofeng@caict.ac.cn](mailto:zhaofeng@caict.ac.cn)

Chongfeng Xie  
China Telecom  
China

Email: [xiechf.bri@chinatelecom.cn](mailto:xiechf.bri@chinatelecom.cn)

Tong Li  
China Unicom  
China

Email: [litong@chinaunicom.cn](mailto:litong@chinaunicom.cn)

Jichun Ma  
China Unicom  
China

Email: [majc16@chinaunicom.cn](mailto:majc16@chinaunicom.cn)

Robbins Mwehaire  
MTN Uganda Ltd.  
Uganda

Email: Robbins.Mwehair@mtn.com

Edmore Chingwena  
MTN Group Limited  
South Africa

Email: Edmore.Chingwena@mtn.com

Shuping Peng  
Huawei Technologies  
China

Email: pengshuping@huawei.com

Zhenbin Li  
Huawei Technologies  
China

Email: lizhenbin@huawei.com

Yaqun Xiao  
Huawei Technologies  
China

Email: xiaoyaqun@huawei.com