

Network Working Group
Internet-Draft
Intended status: Informational
Expires: September 12, 2019

H. Chen
China Telecom
Z. Li
China Mobile
F. Xu
Tencent
Y. Gu
Z. Li
Huawei
March 11, 2019

Network-wide Protocol Monitoring (NPM): Use Cases
draft-chen-npm-use-cases-00

Abstract

As networks continue to scale, we need a coordinated effort for diagnosing control plane health issues in heterogeneous environments. Traditionally, operators developed internal solutions to address the identification and remediation of control plane health issues, but as networks increase in size, speed and dynamicity, new methods and techniques will be required.

This document highlights key network health issues, as well as network planning requirements, identified by leading network operators. It also provides an overview of current art and techniques that are used, but highlights key deficiencies and areas for improvement.

This document proposes a unified management framework for coordinating diagnostics of control plane problems and optimization of network design. Furthermore, it outlines requirements for collecting, storing and analyzing control plane data, to minimise or negate control plane problems that may significantly affect overall network performance and to optimize path/peering/policy planning for meeting application-specific demands.

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 September 12, 2019.

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
1.1. Role of Telemetry	3
1.2. Role of Control Plane Telemetry	4
2. Terminology	5
3. Problem Statement	5
3.1. Network Troubleshooting Challenges	5
3.2. Network Planning Challenges	7
4. Network-wide Protocol Monitoring (NPM)	7
5. NPM Use Cases	9
5.1. Network Troubleshooting Use Cases	9
5.1.1. IS-IS Route Flapping	10
5.1.2. LSDB Synchronization Failure	11
5.1.3. Route Loop	11
5.1.4. Tunnel Set Up Failure	12

5.2. Network Planning Use Cases	12
5.2.1. Route Policy Validation	12
6. Security Considerations	13
7. Contributors	13
8. Acknowledgments	13
9. References	13
Authors' Addresses	16

1. Introduction

Recently, significant effort has been made to evolve control network resources, using management plane enhancements and control of network state via centralized and distributed control plane methods. There is ongoing effort in the diagnosing of forwarding plane performance degradation, using telemetry-based solutions and in-band data plane OAM. However, less emphasis has been applied on the diagnosing and remediation of health problems related to optimal control of network resources, and diagnosing control plane health issues.

The document outlines the existing set of standards-based tools and highlights the lack of capability for addressing control plane monitoring.

1.1. Role of Telemetry

The concept of network telemetry has been proposed to meet the current and future OAM demands, supporting real-time data collection, process, exportation, and analysis, and an architectural framework of existing Telemetry approaches is introduced in [I-D.song-ntf] [I-D.song-ntf]. Network telemetry provides visibility to the network health conditions, and is beneficial for faster network troubleshooting, network OpEx (operating expenditure) reduction, and network optimization. Telemetry can be applied to the data plane, control plane and management plane. There have been various methods proposed for each plane:

- o Management Plane Telemetry: The management plane telemetry focuses on network operational state retrieval and configuration management. SNMP (Simple Network Management Protocol) [RFC1157], NETCONF (Network Configuration Protocol) [RFC6241] and gNMI (gRPC Network Management Interface) [I-D.openconfig-rtgwg-gnmi-spec] are three widely adopted management plane Telemetry approaches. Data consumers can subscribe to specific data stores through SNMP/gRPC/NETCONF.
- o Control Plane Telemetry: The control plane telemetry works on routing protocol monitoring and routing related data retrieval, e.g., topology, route policy, RIB and so on. BGP monitoring

protocol (BMP) [RFC7854] is proposed to monitor BGP sessions and intended to provide a convenient interface for obtaining BGP route views. Data collected using BMP can be further analyzed with big data platforms for network health condition visualization, diagnose and prediction applications.

- o Data Plane Telemetry: The data plane telemetry works on traffic performance measurement and traffic related data retrieval, e.g., latency, jitter, buffer size and so on. For example, In-situ OAM (iOAM) [I-D.brockners-inband-oam-requirements] embeds an instruction header to the user data packets, and collects the requested data and adds it to the user packet at each network node along the forwarding path. Applications such as path verification, SLA (service-level agreement) assurance can be enabled with iOAM.

1.2. Role of Control Plane Telemetry

The above mentioned telemetry approaches may vary in data type and form, including: encapsulation, serialization, transportation, subscription, and data analysis, thus resulting in various applications. With the network operations and maintenance evolving towards automation and intent-driven, higher requirements are set for each plane. Healthy management plane and control plane are essential for high-quality data service provisioning. The visibility of management and control planes' healthiness provides insights for changes in the data plane.

First of all, the running of control protocols aims to provide and guarantee the network connectivity and reachability, which is the foundation of any data service running above it. The monitoring of the control plane detects the healthiness issue in real time so that immediate troubleshooting actions can be taken, and thus mitigating the affect on data services as much as possible.

Secondly, without route analytics, the dynamic nature of IP networking makes it virtually impossible to know at any time point how traffic is traversing the networks. For example, by collecting real-time BGP routes through BMP and correlating them with traffic data retrieved through data plane telemetry, the operator is able to provide both inter-domain and intra-domain traffic optimization.

Finally, the validation and evaluation of route policies is another common appeal from both carriers and OTTs. The difficulty here majorly lies in the precise definition of the correctness of policies. In other words, the policy validation depends largely on the operator's understanding and manual judgement of the current network status instead of formatted and quantitative command executed

at devices. Thus, it demands visualized presentations of how the policies impact the route changes through control plane telemetry so that operators may have direct judgement of the policy correctness. The conventional separated data collections of route policy and route information is not sufficient for the correctness validation of route policy.

Based on discussions with leading operators, this document identifies the challenges and problems that the current control plane telemetry faces and suggests the data collection requirements. The necessity for a Network-wide Protocol Monitoring (NPM) framework is illustrated and conducted through the discussion of specific use cases.

2. Terminology

IGP: Interior Gateway Protocol

IS-IS: Intermediate System to Intermediate System

BGP: Boarder Gateway Protocol

BGP-LS: Boarder Gateway Protocol-Link State

MPLS: Multi-Protocol Label Switching

RSVP-TE: Resource Reservation Protocol-Traffic Engineering

LDP: Label Distribution Protocol

NPM: Network-wide Protocol Monitoring

NPMS: Network Protocol Monitoring System

BMP: BGP Monitoring Protocol

LSP: Link State Packet

SDN: Software Defined Network

IPFIX: Internet Protocol Flow Information Export

3. Problem Statement

3.1. Network Troubleshooting Challenges

According to Huawei 2016 network issue statistics, about 48% issues of the total amount are routing protocol-related, including protocol adjacency/peer set up failure, adjacency/peer flapping, protocol-

related table error. What's more, the routing protocol issues are not standalone, which simultaneously come with anomaly status in data plane, and are finally reflected on poor service quality and user experience.

Existing methods for protocol troubleshooting include CLI, SNMP, Netconf-YANG/gRPC-YANG and vendor-specific/third party tools.

Using CLI to do per-device check provides adequate per device information, but lacks network-wide vision, thus leading to either massive labor/time consumption checking all devices or fail to localize the source. Besides, complex CLI usage (combination and repeat pattern) requires experience from the NOC person.

Management protocols, like SNMP, Netconf/gRPC, provide information already/to be gathered from the network, which reduces operational complexity, but sacrifices data adequacy compared with CLI. Since the above protocols aren't designed specifically for routing troubleshooting, not all the data source required is currently supported for exportation, and the lack of certain data becomes the troubleshooting bottleneck. For example, in an LSP purge abnormal case caused by continuous corrupted LSP, it's useful to collect the corrupted LSP PDUs for root cause analysis. In addition, for the currently supported, as well as to be supported, data source collection, the data synchronization issue, due to export performance difference of various approaches, can be a concern for data correlation. The data collection requirements depend largely on the use cases, and more details are discussed in Section 5.

Some third party OAM tools provide troubleshooting-customized information collection and analysis. For example, Packet Design uses passive listening to collect IS-IS/OSPF/BGP messages to do route analysis for troubleshooting and path optimization. Such passive listening lacks per-device information collection. For example, to detect the existence of a route loop and analyze the root cause, it not only requires the network-wide RIB/FIB collection, but also requires the route policy information that is responsible for the generation of loop issue.

To summarize here, the currently protocols and tools do not provide sufficient data source for routing troubleshooting. There requires new methods or augmented work to existing methods to enhance the control plane data collection and to support more efficient data correlation.

3.2. Network Planning Challenges

The dynamic nature of IP networks, e.g., peer up/down, prefix advertisement, route change, and so on, has great influence on the service provisioning. With the emerging of new network services, such as automated driving systems, AR (Augmented Reality), and so on, network planning is facing new requirements in order to meet the latency, bandwidth and security demands. The requirements can generally break into two perspectives: 1. sufficient and up-to-date routing data collection as the input for network simulation; 2. accurate what-if simulation to evaluate new network planning actions.

Most existing control plane and data plane simulation tools, e.g., Batfish [Batfish], use device configurations to generate a control/data plane. There exists some concerns w.r.t. such simulation method: 1. in a multi-vendor network understanding and translating the configuration files is a non-trivial task for the simulator; 2. the generated control/control plane is not the 100% mirroring of the actual network, and thus resulting in less accurate simulation results. Thus, it requires real-time routing data collection from the on-going network. Currently, BGP routes and peering states are monitored in real-time by using BMP. However, IS-IS/OSPF/MPLS routing data still lacks legitimate and comprehensive monitoring. Here, not only the data coverage, including RIB/FIB, network topology, peering states and so on, but also the data synchronization of various devices should be considered in order to recover a faithful data/control plane within the simulator.

4. Network-wide Protocol Monitoring (NPM)

With the above mentioned challenges facing the control plane telemetry, it is of great value to identify the requirements from typical use cases, and the gaps between the requirements and existing methods. It is thus necessary to propose a comprehensive control plane telemetry framework, as shown in Figure 1.

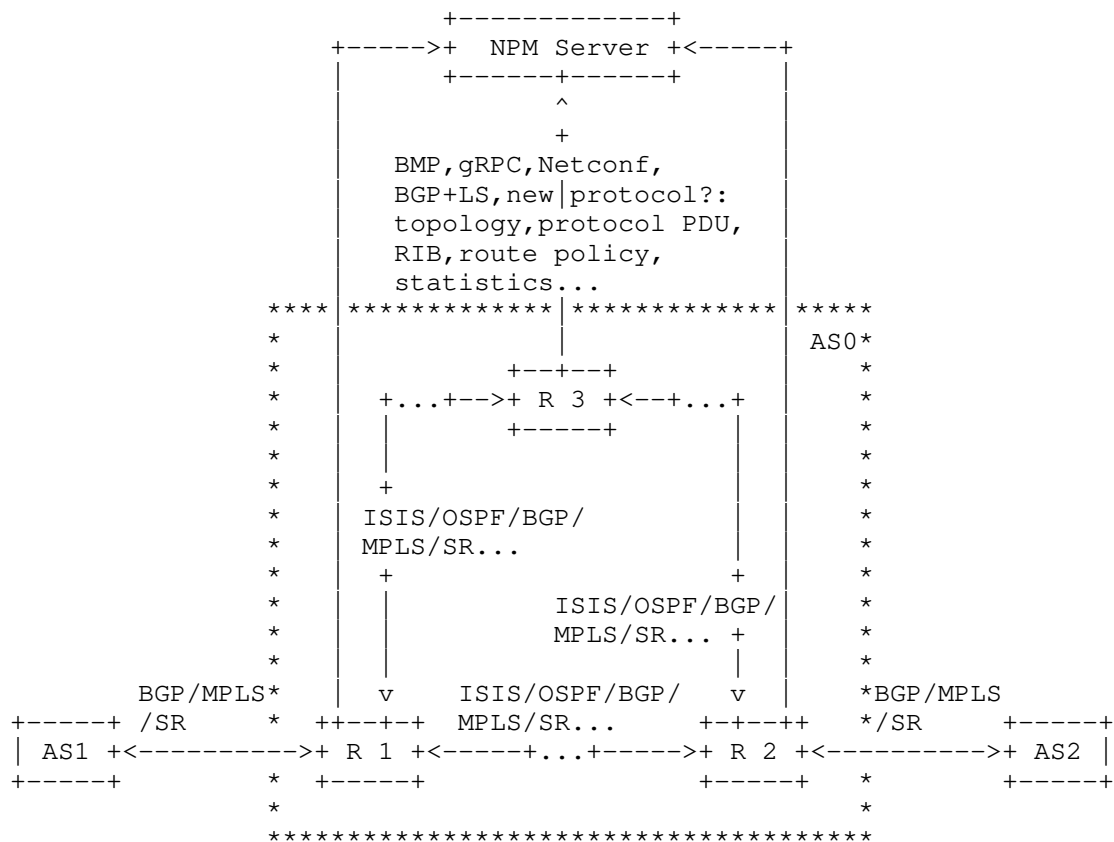


Figure 1: NPM framework

Under the NPM framework, the challenges, use cases, requirements, gaps, and solutions options are to be identified and discussed. The NPM problem space is depicted in Figure 2. Two general requirements are concluded from the challenges discussed above.

- o The requirement of a "tunnel" for the control plane data export: There should be a way (or ways) of exporting the required control plane data, and the export performance (e.g., data modeling, encoding and transmission) should be able to meet per application requirements;
- o The requirement of adequate data collection: In order to support specific troubleshooting and planning use cases, the collected data coverage, including the data type coverage and the network coverage, should be adequate. The data type coverage refers to data such as protocol PDUs, RIBs, policy and so on, and the

network type coverage refers to the devices providing such information.

More specific requirements may vary case by case, but it is a common appeal to guarantee a valid tunnel and adequate data collection.

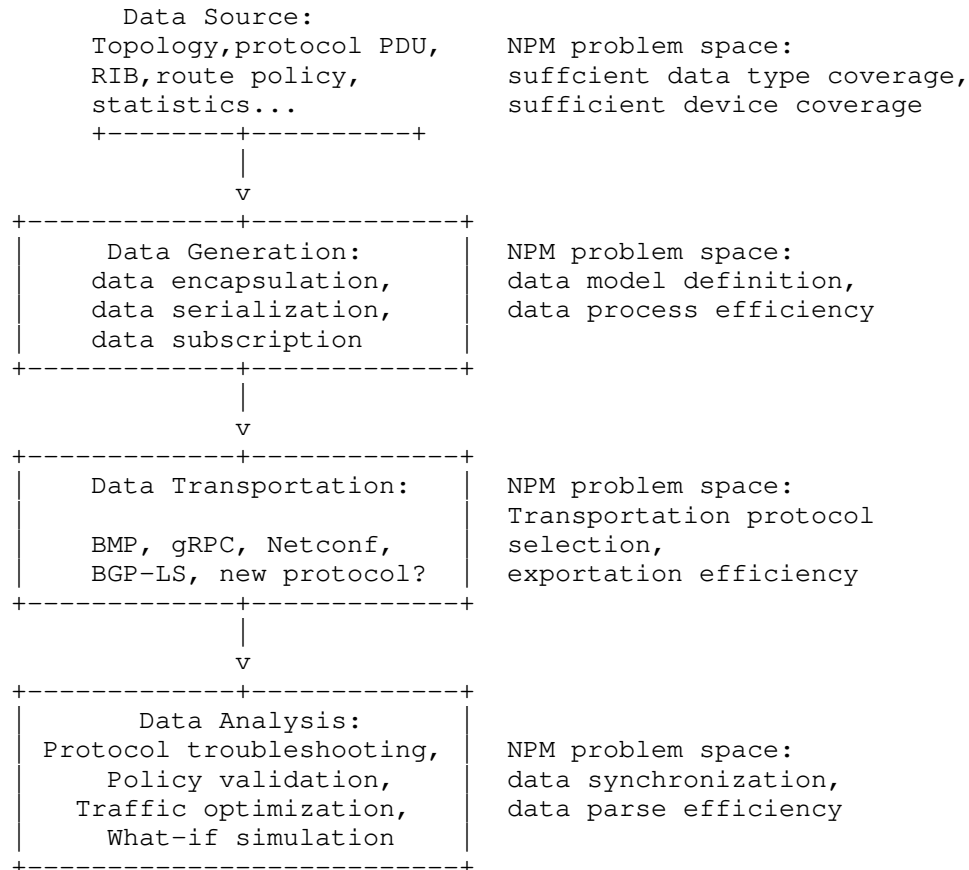


Figure 2: NPM problem space

5. NPM Use Cases

5.1. Network Troubleshooting Use Cases

We have identified several typical routing issues that occur frequently in the network, and are typically hard to localize.

5.1.1. IS-IS Route Flapping

The IS-IS Route Flapping refers to the situation that one or more routes appear and then disappear in the routing table repeatedly. Route flapping usually comes with massive PDUs interactions (e.g., LSP, LSP purge...), which consume excessive network bandwidth, and excessive CPU processing. In addition, the impact is often network-wide. The localizing of the flapping source and the identifying of root causes haven't been easy work due to various reasons.

The flapping can be caused by system ID conflict, IS-IS neighborship flapping, route source flapping (caused by import route policy misconfiguration), device clock dis-function with abnormal LSP purge (e.g., 100 times faster) and so on.

- o The system ID conflict check is a network-wide work. If such information is collected centrally to a controller/server, the issues can be identified in seconds, and more importantly, in advance of the actual flapping event.
- o The IS-IS neighborship flapping is typically caused by interface flapping, BFD flapping, CPU high and so on. Conventionally, to locate the issue, operators typically identify the target device(s), and then log in the devices to check related statistics, parsed protocol PDU data and configurations. The manual check often requires a combination of multiple CLIs (check cost/next hop/exit interface/LSP age...) in a repeated manner, which is time-consuming and requires rich OAM experience. If such statistics and configuration data were collected at the server in real-time, the server may analyze them automatically or semi-automatically with troubleshooting algorithms implemented at the server.
- o In the case that route policies are misconfigured, which then causes the route flapping, it's typically difficult to directly identify the responsible policy in a short time. Thus, if the route change history is recorded in correlation with the route policy, then with such record collected at the server, the server can directly identify the responsible policy with the one-to-one mapping between policy processing and the route attribute change.
- o In the case that flapping comes with abnormal LSP purges, it may be due to continuous LSP corruptions with falsified shorter Remaining Lifetime, or the clock running 100 times faster with 100 times more purge LSPs generated. In order to identify the purge originator, RFC 6232 [RFC6232] proposes to carry the Purge Originator Identification (POI) TLV in IS-IS. However, to analyze

the root cause of such abnormal purges, the collection and analysis of LSP PDUs are needed.

5.1.2. LSDB Synchronization Failure

During the IS-IS flooding, sometimes the LSP synchronization failure happens. The synchronization failure causes can be generally classified into three cases:

- o Case 1, the LSP is not correctly advertised. For example, an LSP sent by Router A fails to be synchronized at Router B. It can be due to incorrect route export policy, or too many prefixes being advertised which exceeds the LSP/MTU threshold, and so on at Router A.
- o Case 2, LSP transmission error, which is typically caused by IS-IS adjacency failure, .e.g., link down/BFD down/authentication failure.
- o Case 3, the LSP is received but not correctly processed. The problem that happens at Router B can be faulty route import policy, or Router B being in Overload mode, or the hardware/software bugs.

With sufficient ISIS PDU related statistics and parsed PDU information recorded at the device, the neighborship failure in Case 2 can be typically diagnosed at Router A or Router B independently. With such diagnosing information collected (e.g., in the format of reason code) in real-time, the server can identify the root synchronization issue with much less time and labor consumption compared with conventional methods. In Case 1 & 3, the failure is mostly caused by incorrect route policy and software/hardware issue. By comparing the LSDB with the sent/received LSP, differences can be recognized. Then the difference may further guide the localization of the root cause. Thus, by collecting the LSDBs and sent/received LSPs from the two affected neighbors, the server can have more insights at the synchronization failure.

5.1.3. Route Loop

Incorrect import policy, such as incorrect protocol priority (distance) or improper default route configuration, may result in a route loop. TTL anomaly report or packet loss complain triggers loop alarm. However, locating the exact device(s) and more importantly the responsible configuration/policy is definitely non-trivial work. The generation of routing information base/forwarding information base (RIB/FIB) is related to various protocols and massive route

policies, which often makes it hard to locate the loop source in a timely manner.

If the network-wide RIB/FIB data can be collected in real-time, the server is able to run loop detection algorithms to detect and locate the loop. More importantly, with real-time RIB/FIB collected as the input for network simulator, loop can be predicted with what-if simulations of network changes, such as new policy, or link failure.

5.1.4. Tunnel Set Up Failure

The MPLS label switch path set up, either using RSVP-TE or LDP, may fail due to various reasons. Typical troubleshooting procedures are to log in the device, and then check if the failure lies on the configuration, or path computation error, or link failure. Sometimes, it requires the check of multiple devices along the tunnel. Certain reason codes can be carried in the Path-Err/ResvErr messages of RSVP-TE, while other data are currently not supported to be transmitted to the path ingress/egress node, such as the authentication failure. In this case, if the tunnel configurations of devices along the tunnel, as well as the link states, and other reasons diagnosed by each device can be collected centrally, the server is able to do a thorough analysis and find the root cause.

5.2. Network Planning Use Cases

Monitoring and analyzing the network routing events not only help identify the root causes of network issues, but also provide visibility of how routing changes affect network traffic. With the benefit of data plane telemetry, such as iOAM and IPFIX, network traffic matrices can be generated to give a glance of the current network performance. More specifically, traffic matrices visualize the current and historical network changes, such as link utilization, link delay, jitter, and so on. While traffic matrices provide "what" are the network changes, the control plane event monitoring, such as adjacency/peering failure, route flapping, prefix advertize/withdraw, provides "why".

5.2.1. Route Policy Validation

Route policy validation has been a great concern for operators when implementing new policies as well as optimizing existing policies. Validation comes in two perspectives:

- o Firstly, there requires valid monitoring of implemented policy correlated with network changes to understand how one policy impacts routing in both single-device and network-wide views. Conventionally, policy/configuration data collection (e.g.,

through Netconf/YANG) is separate from route information collection (e.g., BMP), which lacks correlation between policy and routes. Thus, even with both information at hand, it is still difficult for the operator to figure out how a policy impacts the route change. If the route change is recorded correlated with policy processing, the server can directly identify the impact through the correlation analysis of such data collected from all devices.

- o Secondly, there requires pre-check of policy impact using simulation tools. Most existing simulation tools use device configurations to generate a control plane/data plane, and then run what-if simulations to evaluate a new policy. However, there exists difference between the on-going network and the generated control/data plane, and thus leading the simulation results less effective. If the control/data plane snapshot (e.g., topology, protocol neighbor state, RIB...) of the on going network is realized and taken as the input of the simulation, the reliability of the evaluation can be greatly improved.

6. Security Considerations

TBD

7. Contributors

TBD

8. Acknowledgments

TBD

9. References

[Batfish] etc., A. F., "A General Approach to Network Configuration Analysis", May 2015.

[I-D.brockners-inband-oam-requirements]

Brockners, F., Bhandari, S., Dara, S., Pignataro, C., Gredler, H., Leddy, J., Youell, S., Mozes, D., Mizrahi, T., Lapukhov, P., and r. remy@barefootnetworks.com, "Requirements for In-situ OAM", draft-brockners-inband-oam-requirements-03 (work in progress), March 2017.

- [I-D.ietf-grow-bmp-adj-rib-out]
Evens, T., Bayraktar, S., Lucente, P., Mi, K., and S. Zhuang, "Support for Adj-RIB-Out in BGP Monitoring Protocol (BMP)", draft-ietf-grow-bmp-adj-rib-out-03 (work in progress), December 2018.
- [I-D.ietf-grow-bmp-local-rib]
Evens, T., Bayraktar, S., Bhardwaj, M., and P. Lucente, "Support for Local RIB in BGP Monitoring Protocol (BMP)", draft-ietf-grow-bmp-local-rib-02 (work in progress), September 2018.
- [I-D.ietf-netconf-yang-push]
Clemm, A., Voit, E., Prieto, A., Tripathy, A., Nilsen-Nygaard, E., Bierman, A., and B. Lengyel, "Subscription to YANG Datastores", draft-ietf-netconf-yang-push-22 (work in progress), February 2019.
- [I-D.openconfig-rtgw-gnmi-spec]
Shakir, R., Shaikh, A., Borman, P., Hines, M., Lebsack, C., and C. Morrow, "gRPC Network Management Interface (gNMI)", draft-openconfig-rtgw-gnmi-spec-01 (work in progress), March 2018.
- [I-D.song-ntf]
Song, H., Zhou, T., Li, Z., Fioccola, G., Li, Z., Martinez-Julia, P., Ciavaglia, L., and A. Wang, "Toward a Network Telemetry Framework", draft-song-ntf-02 (work in progress), July 2018.
- [RFC1157] Case, J., Fedor, M., Schoffstall, M., and J. Davin, "Simple Network Management Protocol (SNMP)", RFC 1157, DOI 10.17487/RFC1157, May 1990, <<https://www.rfc-editor.org/info/rfc1157>>.
- [RFC1191] Mogul, J. and S. Deering, "Path MTU discovery", RFC 1191, DOI 10.17487/RFC1191, November 1990, <<https://www.rfc-editor.org/info/rfc1191>>.
- [RFC1195] Callon, R., "Use of OSI IS-IS for routing in TCP/IP and dual environments", RFC 1195, DOI 10.17487/RFC1195, December 1990, <<https://www.rfc-editor.org/info/rfc1195>>.
- [RFC1213] McCloghrie, K. and M. Rose, "Management Information Base for Network Management of TCP/IP-based internets: MIB-II", STD 17, RFC 1213, DOI 10.17487/RFC1213, March 1991, <<https://www.rfc-editor.org/info/rfc1213>>.

- [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>>.
- [RFC3209] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V., and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, DOI 10.17487/RFC3209, December 2001, <<https://www.rfc-editor.org/info/rfc3209>>.
- [RFC3719] Parker, J., Ed., "Recommendations for Interoperable Networks using Intermediate System to Intermediate System (IS-IS)", RFC 3719, DOI 10.17487/RFC3719, February 2004, <<https://www.rfc-editor.org/info/rfc3719>>.
- [RFC3988] Black, B. and K. Kompella, "Maximum Transmission Unit Signalling Extensions for the Label Distribution Protocol", RFC 3988, DOI 10.17487/RFC3988, January 2005, <<https://www.rfc-editor.org/info/rfc3988>>.
- [RFC6232] Wei, F., Qin, Y., Li, Z., Li, T., and J. Dong, "Purge Originator Identification TLV for IS-IS", RFC 6232, DOI 10.17487/RFC6232, May 2011, <<https://www.rfc-editor.org/info/rfc6232>>.
- [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>>.
- [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>>.
- [RFC7854] Scudder, J., Ed., Fernando, R., and S. Stuart, "BGP Monitoring Protocol (BMP)", RFC 7854, DOI 10.17487/RFC7854, June 2016, <<https://www.rfc-editor.org/info/rfc7854>>.
- [RFC8231] Crabbe, E., Minei, I., Medved, J., and R. Varga, "Path Computation Element Communication Protocol (PCEP) Extensions for Stateful PCE", RFC 8231, DOI 10.17487/RFC8231, September 2017, <<https://www.rfc-editor.org/info/rfc8231>>.

Authors' Addresses

Huanan Chen
China Telecom
109 West Zhongshan Ave
Guangzhou
China

Email: chenhuanan@gsta.com

Zhenqiang Li
China Mobile
No. 32 Xuanwumenxi Ave., Xicheng District
Beijing
China

Email: lizhenqiang@chinamobile.com

Feng Xu
Tencent
Guangzhou
China

Email: oliverxu@tencent.com

Yunan Gu
Huawei
156 Beiqing Rd
Beijing
China

Email: guyunan@huawei.com

Zhenbin Li
Huawei
156 Beiqing Rd
Beijing
China

Email: lizhenbin@huawei.com

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

S. Homma
H. Nishihara
NTT
T. Miyasaka
KDDI Research
A. Galis
University College London
V. Ram OV
Independent Research Consultant India
D. Lopez
LM. Contreras
J. Ordonez-Lucena
Telefonica I+D
P. Martinez-Julia
NICT
L. Qiang
Huawei Technologies
R. Rokui
L. Ciavaglia
Nokia
X. de Foy
InterDigital Inc.
November 3, 2019

Network Slice Provision Models
draft-homma-slice-provision-models-02

Abstract

Network slicing is an approach to provide separate virtual network based on service requirements, and it's a key feature of the 5G. 3GPP has standardized the specifications for network slicing in the 5GS, but there are still some problems for realization of end-to-end network slices. For complementing the lacks or expanding the usability, several organizations are proceeding standardization. However, the definitions and scopes of network slicing vary to some degree from one organization to another. This document provides classification of provisioning models of network slice for clarifying the differences on the definitions and scopes.

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 and Motivation	3
1.1. Roles in Network Slice Provisioning	3
1.1.1. Definitions in 3GPP on NS Provision Roles	4
1.2. High-level Problem Statement	5
2. Definition of Terms	5
3. General Requirements for Network Slicing	8
4. Network Slice Structure	8
4.1. Resources for Structuring Network Slices	9
4.2. Basic Network Slice Structure	12
4.3. Stakeholders in the Structuring Network Slices	15
5. Variations of Network Slice Creation	15
5.1. Ready-made Network Slice	16
5.2. Custom-made Network Slice	16
5.3. semi-Custom-made Network Slice	16
6. Network Slice Provision Models	16
6.1. Categorization of NS Provision Models	16
6.1.1. SaaS-like Model	17
6.1.2. PaaS-like Model	17
6.1.3. IaaS-like Model	17
6.2. Mapping of NS Provision Models and Infrastructure Layering	18

6.3. Configurable Parameters/Attributes for NS	20
6.4. Capability of NS Tenant on each Provision Model	20
6.4.1. Capability in SaaS-like Model	21
6.4.2. Capability in PaaS-like Model	21
6.4.3. Capability in IaaS-like Model	21
7. Security Considerations	21
8. IANA Considerations	22
9. Acknowledgement	22
10. Informative References	22
Appendix A. NS Structure in the 3GPP 5GS	23
Authors' Addresses	24

1. Introduction and Motivation

Network slicing is an approach to provide separate virtual networks depending on requirements of each service. Network slicing receives attention due to factors such as diversity of services and devices, and it is also a fundamental concept of the 5G for applying networks to such various types of requirements.

In addition, network slicing is expected to enable a business model to provide dedicated logical networks to 3rd parties or vertical customers on-demand, called NSaaS (Network Slice as a Service). For such usage, in network slicing, provision of networks able to guarantee communication characteristics end to end would be required. However, the definitions are not harmonized over several SDOs (Standards Developing Organizations).

This document clarifies provision patterns of network slice, and provides the definitions and scope of network slicing which are available over several organizations. Furthermore, the deliverables would be help for evaluating applicability of existing technologies/solutions to network slicing.

1.1. Roles in Network Slice Provisioning

The widespread of system and network virtualization technologies has conducted to new business opportunities, enlarging the offer of IT resources in the form of Network Slices (NS). As a consequence, there is a clear differentiation between the owner of physical resources, the infrastructure operator, and the intermediary that conforms and delivers network services to the final customers, the Virtual Network Operator (VNO).

VNOs aim to exploit the virtualized infrastructures to deliver new and improved services to their customers. However, current NS techniques offer poor support for VNOs to control their resources. It has been considered that the infrastructure operator is

responsible of the reliability of the NS elements but several situations advocate the VNO to gain a finer control on its resources. For instance, dynamic events, such as the identification of new requirements or the detection of incidents within the virtual system, might urge a VNO to quickly reform its virtual infrastructure and resource allocation. However, the interfaces offered by current virtualization platforms do not offer the necessary functions for VNOs to perform the elastic adaptations they require to tackle with their dynamic operation environments.

1.1.1. Definitions in 3GPP on NS Provision Roles

3GPP has defined multiple roles related to 5G networks and network slicing management in [TS.28.540-3GPP] and they include:

- o Communication Service Customer (CSC): Uses communication services, e.g. end user, tenant, vertical
- o Communication Service Provider (CSP): Provides communication services. Designs, builds and operates its communication services. The CSP provided communication service can be built with or without network slice.
- o Network Operator (NOP): Provides network services. Designs, builds and operates its networks to offer such services.
- o Network Equipment Provider (NEP): Supplies network equipment to network. For sake of simplicity, VNF Supplier is considered here as a type of Network Equipment Provider. This can be provided also in the form of one or more appropriate VNF(s).
- o Virtualization Infrastructure Service Provider (VISP): Provides virtualized infrastructure services. Designs, builds and operates its virtualization infrastructure(s). Virtualization Infrastructure Service Providers may also offer their virtualized infrastructure services to other types of customers including to Communication Service Providers directly, i.e. without going through the Network Operator.
- o Data Centre Service Provider (DCSP): Provides data centre services. Designs, builds and operates its data centres.
- o NFVI Supplier: Supplies network function virtualization infrastructure to its customers.
- o Hardware Supplier: Supplies hardware.

In case of Network Slice as a Service (NSaaS), the Communication Service Provider (CSP) role can be refined into Network Slice Provider (NSP). The Communication Service Customer (CSC) role can be refined into Network Slice Customer (NSC). A NSC can, in turn, offer its own communication services to its own customers, being thus CSP at the same time.

1.2. High-level Problem Statement

Beyond their heterogeneity, which can be resolved by software adapters, NS platforms do not offer common methods and functions, so it is difficult for the virtual network controllers used by the VNOs to actually manage and control virtual resources instantiated on different platforms, not even considering different infrastructure operators. Therefore, it is necessary to reach a common definition of the functions that should be offered by underlying platforms to enable such overlay controllers with the possibility of allocate and deallocate resources dynamically and get monitoring data about them.

Such common methods should be offered by all underlying controllers, regardless of being network-oriented (e.g., ODL, ONOS, Ryu) or computing-oriented (e.g., OpenStack, OpenNebula, Eucalyptus). Furthermore, it is also important for those platforms to offer some "PUSH" function to report resource state, avoiding the need for the VNO's controller to "POLL" for such data. A starting point to get proper notifications within current REST APIs could be to consider the protocol proposed by the [WEBPUSH-WG].

Finally, in order to establish a proper order and allow the coexistence and collaboration of different systems, a common ontology regarding network and system virtualization should be defined and agreed, so different and heterogeneous systems can understand each other without requiring to rely on specific adaptation mechanisms that might break with any update on any side of the relation.

2. Definition of Terms

This section lists definitions and terms related to network slicing. This document refers terms and view points on network slicing in some SDOs, such as 3GPP([TS.23.501-3GPP], [TS.28.530-3GPP], [TS.28.540-3GPP], [TR.28.801-3GPP]and [TR.28.804-3GPP]), and NGMN ([NGMN-5G-White-Paper]). However the scope of this document is not network slicing which is mobile specific but one for general networks, and thus some of definitions in this document may be different from ones of those documents.

Network Slicing: Network slicing indicates a technology, an approach, or a concept to create logical separate 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. In 3GPP definitions, an NS potentially includes both data plane and control plane resources/functions. An NS can have multiple network slice subnets (Radio Access Network/RAN, Transport Network/TN, Core Network/CN, etc.).

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: A Network Slice Subnet is a representation of a set of required resources. It is composed and managed as a group of network elements.

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 a virtual network connecting among end points. It is composed of one or multiple NSSIs. This term is original of this document and is used when it should be emphasized that the target NSI provides connectivity from end to end. As an example, for providing an E2E-NSI on the 3GPP 5G network, combining three types of NSIs: RAN-, TRN-, and CN-NSIs would be required.

Transport Network(TN)-NSSI: A set of connections between various network functions (VNF or PNF) with deterministic SLAs. They can be implemented (aka realized) with various technologies (e.g. IP, Optics, FN, Microwave) and various transport (e.g. RSVP, Segment routing, ODU, OCH etc). The overview of NSI composed with TRN-NSSI is shown in Appendix A.

Radio Access Network(RAN)-NSSI: Regardless of RAN deployment (e.g. distributed-RAN, Centralized-RAN or Cloud-RAN, a RAN-NSSI creates a dedicated and logical resource on RAN for each NSI which are completely. The overview of NSI composed with RAN-NSSI is shown in Appendix A.

Core Network(CN)-NSSI: Regardless of Core deployment, a CN-NSI creates a dedicate and logical resource on Core network for each NSI which are completely. The overview of NSI composed with CN-NSSI is shown in Appendix A.

Network Slice as a Service (NSaaS): An NSaaS is a service delivery model in which a third-party provider or a vertical customer hosts NSIs and makes them available to customers. In this model, there 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 NSIs from NS providers.

Network Slice Stakeholder (NS Stakeholder): An NS stakeholder is an actor in network slicing, and has roles of either NS provider or tenant.

Infrastructure Operator: An infrastructure operator is an organization who manages infrastructure networks or data centers for running NSIs. In the most of cases, infrastructure operators are initial NS providers on NSaaS. Also, some of them may be NS tenants simultaneously.

Vertical Customer: A vertical customer is a organization who provides some communicating services with using NSIs on NSaaS model. In many cases, a vertical customer become the final NS tenant on NSaaS. For example, video gaming companies or vehicle vendors will possibly be vertical customers.

Virtual Network Operator (VNO): A VNO is a person or group that operates virtual networks composed with resources or NSSIs rent from infrastructure operators and provides such virtual networks as NSIs to vertical customers who are final NS tenants. In some cases, infrastructure operators have this role in addition to operating their own infrastructure simultaneously.

Domain: A domain is a group of a network and devices administrated under a policy-based common set of rules and procedures.

Resource: A resource is an element used to create virtual networks. There are several types of resources, i.e., connectivity, computing and storage. The details are described Section 4.1

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 VNFs run as different software and processes on top of industry-standard high-volume servers, switches and storage, or cloud computing infrastructure. They are capable of implementing network functions traditionally implemented via custom hardware appliances and middleboxes (e.g., router, NAT, firewall, load balancer, etc.).

Network Operation System: A network operation system is an entity or a group of entities for operating network nodes and functions as compositions of infrastructure network. For example, OSS/BSS, orchestrator, and EMS are considered to be network operation systems.

3. General Requirements for Network Slicing

On network slice operations, capabilities for dynamic instantiation, change, and deletion should be required because an NSI is established based on received orders from tenants in NSaaS. From this aspect, some mechanisms to design a network based on service requirements and to convert those to concrete configurations based on the design would be required.

In addition, each NS has to maintain concrete communication characteristics end to end, and resource reservations on data plane and isolation among NSIs would be required. Isolation is a concept to prevent the reduction of communication quality caused by disturbance from other NSs, and it may have some levels of enforcement, such as hard or soft isolations. In some cases, for providing appropriate communication between client and server, it would be allowed for NS tenants to put their applications as contents server on NSIs by using computing resources.

The required agility of slice operation and granularity of end to end communication quality requested can vary depending on provision model.

4. Network Slice Structure

This section describes resources used for structuring NSs and the basic structure of E2E-NS.

4.1. Resources for Structuring Network Slices

A network slice is structured as combinations of the resources it uses. Such resources are mainly categorized into three classes: network/WAN, computing/NFVI, and functionality resources. Variations of each resources are described below. (Note that the lists are not exhaustive.)

Network (WAN) Resources:

- * Connectivity:
 - + (v)Link
 - Bandwidth per link/session
 - Connected area/end points
 - Forwarding route/path (e.g., for traffic engineering, redundancy)
 - Communication Priority (e.g., QoS class)
 - Range of jitter amount
 - + Interface of vNode
 - QoS setting (e.g., Queue size, DSCP remarking, PIR/CIR)
 - Filter setting
 - + vRouter/vSwitch (# Treated as a set of (v)links and interfaces of vNodes.)
- * Multicast support
- * Encryption support
- * Authentication support
- * Metadata conveyance (e.g., subscriber ID)
- * Protocols for slice data plane:
 - + VLAN
 - + IPoE (IPv4 or IPv6)

- + MAP-E
- + DS-Lite
- + PPPoE
- + L2TP
- + GRE
- + MPLS
- + VxLAN
- + Geneve
- + GTP-U
- + Segment Routing MPLS
- + Segment Routing IPv6
- + NSH
- + FlexE
- + Other

Computing(NFVI) Resources:

- * (v)CPU core
- * Storage
- * Memory
- * Disk
- * vNIC
- * Connectivity to VNF instances
- * Virtual Deployment Unit:
 - + Virtual Machine (VM)
 - + container

- + micro kernel

- * Resource Deployment Location (i.e., edge DC, central DC, public cloud, ..., etc.)

Functionality Resources:

- * Image:

- + Data Plane (DP) NF:

- GateWay (GW) function:

- o Access Point Type (e.g., for radio, Wi-Fi, and fixed accesses)
 - o Slice Selection Setting
 - o Terminate protocol
 - o Authentication

- Security Appliance:

- o IPS (Intrusion Prevention System)
 - o IDS (Intrusion Detection System)
 - o WAF (Web Application Firewall)

- DPI

- Load Balancer

- TCP Accelerator

- Video Optimizer

- Parental Control

- Mobile DP functions (Ref. 3GPP 5GS)

- gNB

- UPF

- Uplink Classifier

- + Control Plane (CP) NF:
 - DHCP
 - o Fixed IP address allocation
 - o Dynamic IP address allocation
 - o The number of registered devices
 - DNS
 - VoIP (SBC, SIP server)
 - Mobile CP function (Ref. 3GPP 5GS)
 - o AMF (Access and Mobility management Function)
 - o SMF (Session Management Function)
 - o PCF (Policy Control Function)
 - o UDM (Unified Data Management)
 - o NEF (Network Exposure Function)
- * Provided VNF Type (e.g., open source, product of vender#A, ..., etc.)
- * Function location (e.g., edge DC, central DC, Public cloud, etc.)

In terms of security or usability for NS tenants, some abstraction on resource information would be required, however both setting parameters of underlay infrastructure and abstracted information may coexist in these lists.

For abstraction of parameters of underlay networks, some additional protocols or functions (like [RFC8453]) would be required. Moreover, for providing strict communication qualities, combinations of some technologies may be useful (ref. [I-D.dong-teas-enhanced-vpn]).

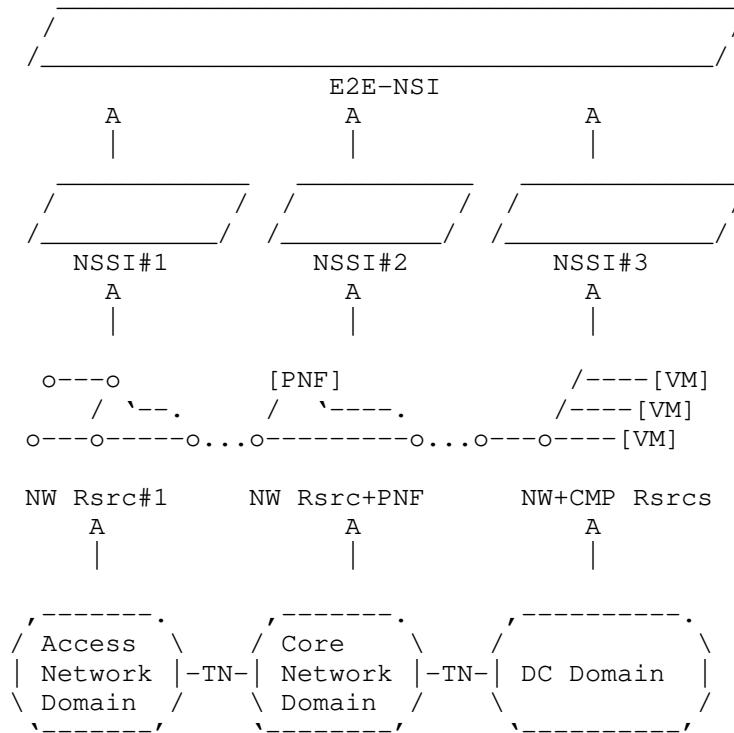
4.2. Basic Network Slice Structure

An E2E-NSI is constructed by stitching NSSIs instantiated on each participating domain. This includes the simplest case of a single

NSSI as an E2E NS. Domain types where some NSSIs are established are described below:

- o Fixed access network
- o Mobile access network
- o Transport network
- o Fixed core network
- o Mobile core network
- o Data center (DC)
 - * Edge DC
 - * Central DC
- o Private network
 - * Enterprise
 - * Factory
 - * Utilities
 - * Farming
 - * Home/SOHO
 - * Other

Figure 1 describes the overview of this structure. Resources in each domain (e.g., access, core networks, and DC) are handled by management entities and constitute an NSSI. An E2E-NSI is established by stitching these NSSIs. Ways to stitch NS-subnets are described in [I-D.defoy-coms-subnet-interconnection] and [I-D.homma-nfvrg-slice-gateway].



*Legends

NW Rsrc : Network Resource

CMP Rsrc: Computing Resource

o : virtual/physical node structuring NSI

-- : virtual/physical link structuring NSI

-TN-: Transport Network

[PNF]: Physical Network Function Appliance on NSI

[VM] : Virtual Machine Instance on NSI

Figure 1: Overview of NS Structure

Although it is shown that an NSSI belongs to just only one E2E-NSI in Figure 1, it may be allowed that multiple E2E-NSIs share an NSSI. Some resources may belong to multiple NSSI as well.

In addition, structure on composition of NSI may be recursive. In other words, even though Figure 1 shows a case where NSSIs compose directly an E2E-NSI, in some cases, NSSIs compose an NSI which is a part of an E2E-NSI. The overview is shown in Figure 2. In this

figure, NSI#4 is composed of NSSI#1 and NSSI#2, and it structures E2E-NSI#5 with NSSI#3.

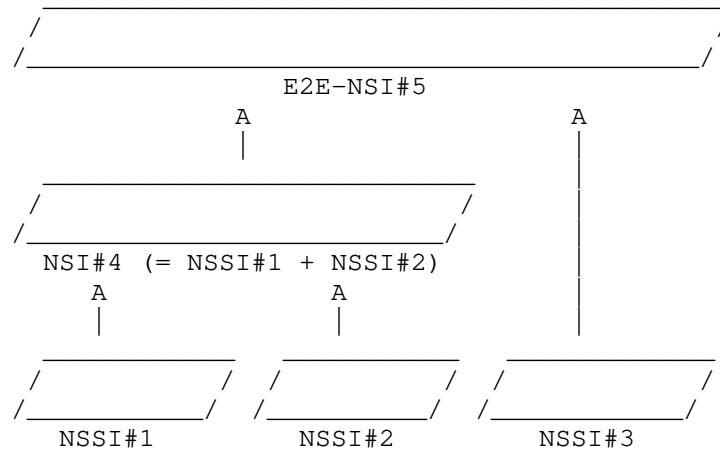


Figure 2: Overview of NS recursive structure

4.3. Stakeholders in the Structuring Network Slices

Potential stakeholders in network slicing are described below:

- o NSSI provider: infrastructure operator
- o Intermediate-NSI provider: infrastructure operator, VNO
- o E2E-NSI provider: infrastructure operator, VNO, service provider
- o NS tenant: infrastructure operator, VNO, service provider, enterprise, mass user
- o End customer: enterprise, mass user, etc.

5. Variations of Network Slice Creation

NSs can be classified according to their creation pattern into two types: ready-made(RM) NS, custom-made(CM), and semi-custom-made(sCM) NS. This section describes the features of these types.

5.1. Ready-made Network Slice

RM-NS is an NS creation pattern in which an infrastructure operator decides service requirements by itself, and established based on the requirements in advance. NS tenants select one of RM-NSs whose features are closer to their requirements.

This model doesn't need immediacy on designing of NSI and enables to mitigate the difficulty of implementation compared with other models.

5.2. Custom-made Network Slice

CM-NS is an NS creation pattern in which an NS is established based on an order from a tenant and is provided to it. As examples of usage of CM-NS, an enterprise builds and operates a virtual private network for connecting several bases, or OTT (Over The Top) or other industrial service providers create NSs based on their own requirements and use them as a part of their own services (e.g., connected vehicles/drones, online video games, or remote surgery).

In this model, network operation system would be required to have incorporate intelligence for designing appropriate NSs on-demand.

5.3. semi-Custom-made Network Slice

sCM-NS is a derivation of a CM-NS. In sCM-NS, an NS provider designs the outline of NSs in advance, and a tenant tunes an NS with deciding some parameters or applications run on resources. For example, an infrastructure operator designs a logical network presenting connectivity, and tenants install their own applications on servers running on the logical network.

6. Network Slice Provision Models

This document classifies NS provision models into three categories defined in the following section. The capabilities which NS tenants can have on management of NSs would vary depending on the selected provision model.

6.1. Categorization of NS Provision Models

The provision models are categorized into three models: SaaS (Software as a Service) -like Model, PaaS (Platform as a Service) -like Model, and IaaS (Infrastructure as a Service) -like Model.

6.1.1. SaaS-like Model

In SaaS-like Model, underlay infrastructure is hidden from tenants, and tenants can receive desired communication environment without deep knowledge about network and servers. An NS tenant decides attribute values of its NS, such as bandwidth or latency, based on their requirements, and NS providers design and create NSIs which fulfill the values.

NS tenants need not to grasp detailed configurations in underlay networks in this model. However, it may not be possible to provide strictly desired NS to tenants because of abstraction of configurable parameters. Moreover, it may cause complexity on designing NS catalog due to quantities of selected attributes.

6.1.2. PaaS-like Model

In PaaS-like Model, an NS is represented with several components such as nodes and connectivities among them. An NS tenant can design and customize its desired NS with combining such components. NS providers breakdown the NS designed by the NS tenant to concrete configurations of their infrastructure, and create/change NSSIs by inputting the configurations. An NS tenant is also able to incorporate its own functions or applications into its NSI by using computing resources provided from NS providers.

This model potentially has high customizability of NS rather than SaaS-like model, but needs NS tenants to have some knowledge about network management. In terms of designing NS, the tenants provide outline of their NSs, and thus it would make creation of concrete configurations be easier.

6.1.3. IaaS-like Model

In IaaS-like model, an NS is represented with concrete configurations of underlay infrastructure. NS tenants are able to structure or change their desired NS by controlling infrastructure resources directly.

This model potentially has high customizability of NS rather than other models, but needs NS tenants to have deep knowledge about network and server operation. Also, NS providers need not to recognize NSs on their infrastructure because NS tenants directly manage their NS. Meanwhile, in terms of security and prevention of disturbances among NSs, some limitations on expositions of resources to tenants would be needed.

6.2. Mapping of NS Provision Models and Infrastructure Layering

An example of mapping of each NS provision model is shown in Figure 3.

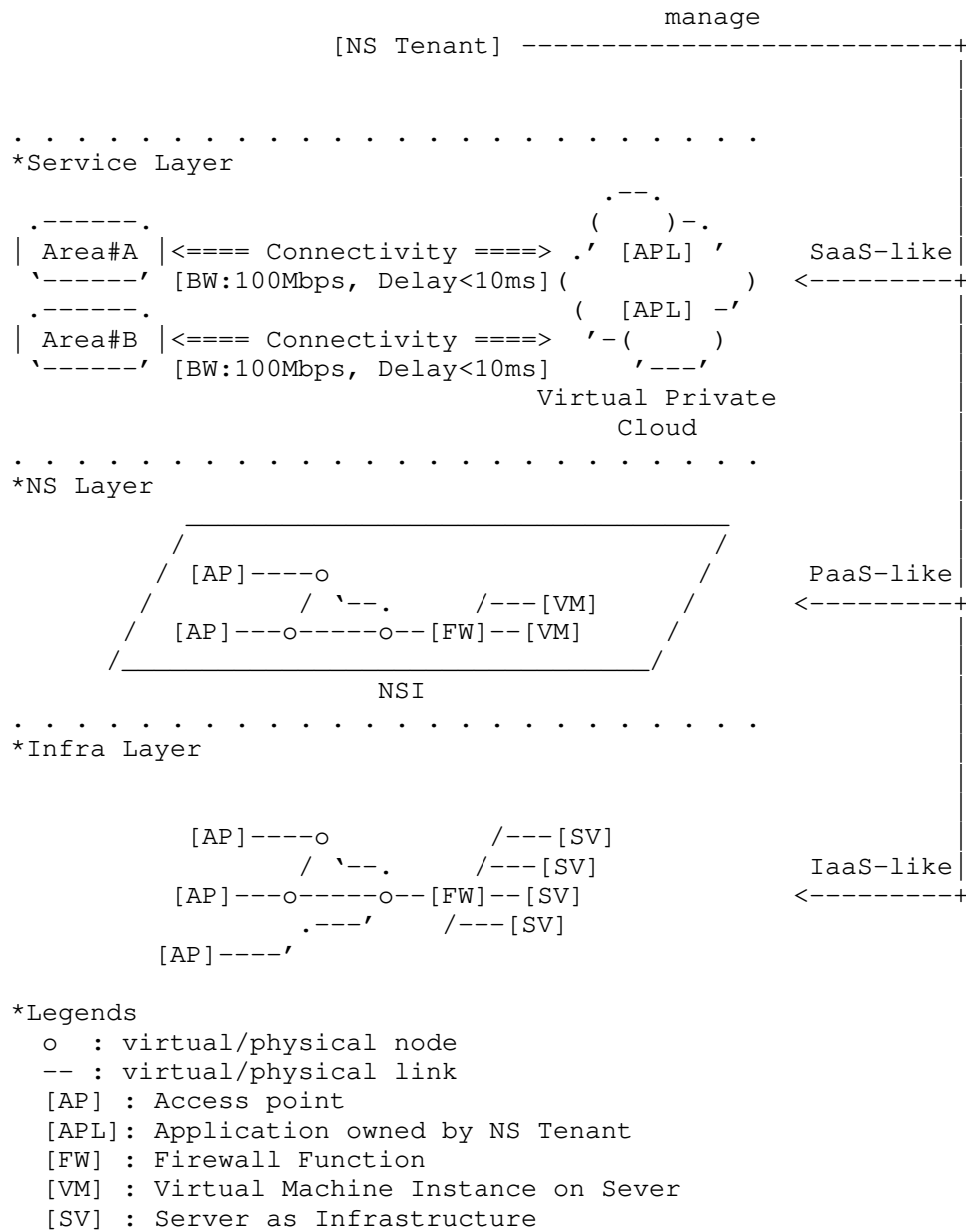


Figure 3: Mapping of NS provision models

In some cases, NSIs provided based on IaaS- or PaaS-like models are coordinated to a form of SaaS-like model by an NS broker , and the NS broker or by the tenant, becoming a NS provider in a recursive manner. For example, a vertical customer sends its high-level requirements to an NS broker create an appropriate NSI with resources provided by infrastructure operators.

6.3. Configurable Parameters/Attributes for NS

In the NS creating procedure, configuration parameters are decided based on requirements which the intended service has. Such service requirements are expressible in the following attribute values.

- o Attributes for Network Resource
 - * bandwidth
 - * latency
 - * jitter
 - * packet loss rate
 - * reliability (e.g., MTBF, MTTF)
- o Attributes for Functionalities Resources
 - * function type (e.g., security, parental control)
 - * throughput
 - * packet error rate
 - * availability

Configurable parameters of components in underlay infrastructure will vary depending on the implementation and structure. Controlled resource types are described in Section 4.1.

6.4. Capability of NS Tenant on each Provision Model

Capability of NS tenants on NS management would vary depending on the NS provision model. This section describes clarification about such capability in each model.

6.4.1. Capability in SaaS-like Model

In SaaS-like Model, an NS is represented for a tenant with attributes values listed in Section 6.3. In other words, an NS tenant never know the concrete configurations of components in underlay infrastructure.

An NS tenant chooses a value from the range presented by the NS provider in each attribute. The NS provider creates or changes a NS by configuraing components in underlay infrastructures based on the decided attribute values.

In terms of telemetry for assurance of service qualities on a NS, a tenant can obtain telemetry information with unit of NSI, and never know ones of underlay components structuring the NS.

6.4.2. Capability in PaaS-like Model

In PaaS-like model, an NS is represented with NF nodes and their connectivities. An NS tenant can indicate functionalities of NF nodes and thier locations. Also, the tenant decides attribute values of connectivities. An NS provider creates or changes an NSI by configuring underlay nodes and links depending on the indication of the tenant. An NS tenant is also able to deploy its own NF as software with provided computing resources.

In terms of telemetry, an NS tenant can obtain telemetry information of NF nodes and connectivities structuring an NS, in addition to whole of NSI.

6.4.3. Capability in IaaS-like Model

In IaaS-like Model, an NS is represented with configurations of (virtual) nodes and (virtual) links connecting the nodes. An NS tenant is able to configure nodes and links in underlay infrastructure. In short, an NS tenant directly design detailed NS and manages it. In addition, an NS tenant inserts its own functions or applications in the NS with using computing resources.

In terms of telemetry, an NS tenant can obtain telemetry information of nodes and links in addition of whole of NSI.

7. Security Considerations

In NSaaS, parts of controls of infrastructures are opened to externals, and thus some mechanisms, such as authentication for APIs, to prevent illegal access would be required.

Other considerations are TBD

8. IANA Considerations

This memo includes no request to IANA.

9. Acknowledgement

The author would like to thank Toru Okugawa for his kind review and valuable feedback.

10. 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.dong-teas-enhanced-vpn]

Dong, J., Bryant, S., Li, Z., Miyasaka, T., and Y. Lee, "A Framework for Enhanced Virtual Private Networks (VPN+) Service", draft-dong-teas-enhanced-vpn-03 (work in progress), November 2018.

[I-D.homma-nfvrg-slice-gateway]

Homma, S., Foy, X., and A. Galis, "Gateway Function for Network Slicing", draft-homma-nfvrg-slice-gateway-00 (work in progress), July 2018.

[NGMN-5G-White-Paper]

NGMN, "NGMN 5G White Paper", February 2015, <<https://www.ngmn.org/5g-white-paper/5g-white-paper.html>>.

[RFC8453] Ceccarelli, D., Ed. and Y. Lee, Ed., "Framework for Abstraction and Control of TE Networks (ACTN)", RFC 8453, DOI 10.17487/RFC8453, August 2018, <<https://www.rfc-editor.org/info/rfc8453>>.

[TR.28.801-3GPP]

3rd Generation Partnership Project (3GPP), "3GPP TR 28.801 (V15.1.0): Study on Management and Orchestration of Network Slicing for next generation network (Release 15)", June 2018, <http://www.3gpp.org/ftp//Specs/archive/28_series/28.801/28801-f10.zip>.

[TR.28.804-3GPP]

3rd Generation Partnership Project (3GPP), "3GPP TR 28.804 (V16.1.0): Study on tenancy concept in 5G networks and network slicing management (Release 16)", October 2019, <http://www.3gpp.org/ftp//Specs/archive/28_series/28.801/28801-f10.zip>.

[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>.

[TS.28.530-3GPP]

3rd Generation Partnership Project (3GPP), "3GPP TS 28.530 (V1.0.0): Management and orchestration of networks and network slicing; Concepts, use cases and requirements (work in progress)", June 2018, <http://ftp.3gpp.org//Specs/archive/28_series/28.530/28530-100.zip>.

[TS.28.540-3GPP]

3rd Generation Partnership Project (3GPP), "3GPP TS 28.540 (V16.0.0): 5G Network Resource Model (NRM); Stage 1 (Release 16)", June 2019, <http://www.3gpp.org/ftp//Specs/archive/28_series/28.540/28540-g00.zip>.

[TS.28.541-3GPP]

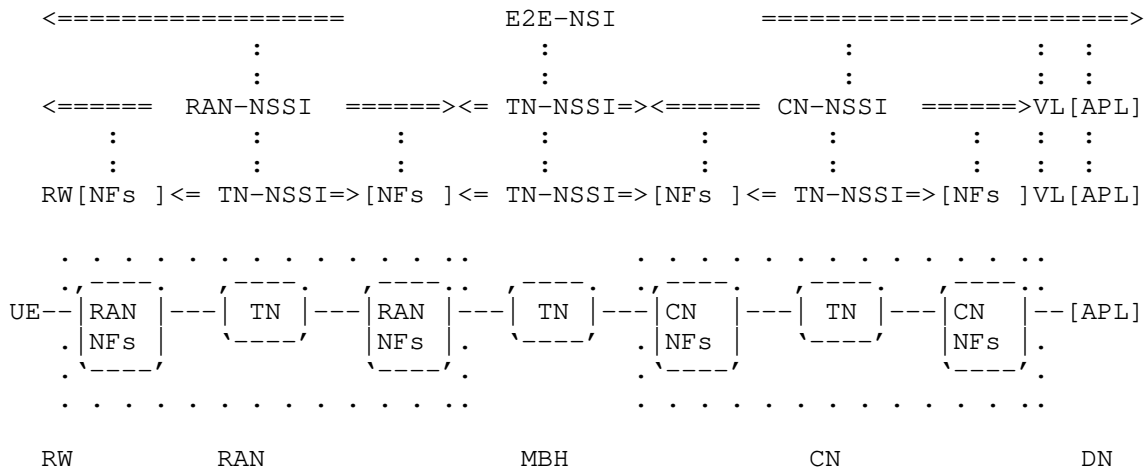
3rd Generation Partnership Project (3GPP), "3GPP TS 28.541 (V15.1.0): 5G Network Resource Model (NRM); Stage 2 and stage 3 (Release 15)", June 2018, <http://www.3gpp.org/ftp//Specs/archive/28_series/28.541/28541-f01.zip>.

[WEBPUSH-WG]

IETF, "Web-Based Push Notifications (webpush)", <<https://datatracker.ietf.org/wg/webpush/about/>>.

Appendix A. NS Structure in the 3GPP 5GS

The overview of structure of NS in the 3GPP 5GS is shown in Figure 4. The terms are described in the 3GPP documents (e.g., [TS.23.501-3GPP] and [TS.28.530-3GPP]).



*Legends

UE: User Equipment
 RAN: Radio Access Network
 CN: Core Network
 DN: Data Network
 TN: Transport Network
 MBH: Mobile Backhaul
 RW: Radio Wave
 NF: Network Function
 APL: Application Server

Figure 4: Overview of Structure of NS in 3GPP 5GS

Authors' Addresses

Shunsuke Homma
 NTT
 Japan

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

Hidetaka Nishihara
 NTT
 Japan

Email: nishihara.hidetaka@lab.ntt.co.jp

Takuya Miyasaka
KDDI Research
Japan

Email: ta-miyasaka@kddi-research.jp

Alex Galis
University College London
UK

Email: a.galis@ucl.ac.uk

Vishnu Ram OV
Independent Research Consultant India
India

Email: vishnu.u@ieee.org

Diego R. Lopez
Telefonica I+D
Spain

Email: diego.r.lopez@telefonica.com

Luis M. Contreras
Telefonica I+D
Spain

Email: luismiguel.contrerasmurillo@telefonica.com

Jose A. Ordonez-Lucena
Telefonica I+D
Spain

Email: joseantonio.ordonezlucena@telefonica.com

Pedro Martinez-Julia
NICT
Japan

Email: pedro@nict.go.jp

Li Qiang
Huawei Technologies
China

Email: qiangli3@huawei.com

Reza Rokui
Nokia
Canada

Email: reza.rokui@nokia.com

Laurent Ciavaglia
Nokia
France

Email: Laurent.ciavaglia@nokia.com

Xavier de Foy
InterDigital Inc.
Canada

Email: Xavier.Defoy@InterDigital.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

Bo Wu (editor)
Huawei

Email: lane.wubo@huawei.com

Robert Wilton (editor)
Cisco Systems

Email: rwilton@cisco.com

Xiaojian Ding

Email: wjswsl@163.com

Network Working Group
Internet-Draft
Intended status: Informational
Expires: 21 October 2022

F. L. Templin, Ed.
G. Saccone
Boeing Research & Technology
G. Dawra
LinkedIn
A. Lindem
V. Moreno
Cisco Systems, Inc.
19 April 2022

A Simple BGP-based Mobile Routing System for the Aeronautical
Telecommunications Network
draft-ietf-rtgwg-atn-bgp-17

Abstract

The International Civil Aviation Organization (ICAO) is investigating mobile routing solutions for a worldwide Aeronautical Telecommunications Network with Internet Protocol Services (ATN/IPS). The ATN/IPS will eventually replace existing communication services with an IP-based service supporting pervasive Air Traffic Management (ATM) for Air Traffic Controllers (ATC), Airline Operations Controllers (AOC), and all commercial aircraft worldwide. This informational document describes a simple and extensible mobile routing service based on industry-standard BGP to address the ATN/IPS requirements.

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 21 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	2
2. Terminology	7
3. ATN/IPS Routing System	9
4. ATN/IPS (Radio) Access Network (ANET) Model	14
5. ATN/IPS Route Optimization	16
6. BGP Protocol Considerations	19
7. Stub AS Mobile Routing Services	21
8. Implementation Status	21
9. IANA Considerations	21
10. Security Considerations	21
10.1. Public Key Infrastructure (PKI) Considerations	22
11. Acknowledgements	23
12. References	23
12.1. Normative References	23
12.2. Informative References	24
Appendix A. BGP Convergence Considerations	26
Appendix B. Change Log	26
Authors' Addresses	27

1. Introduction

The worldwide Air Traffic Management (ATM) system today uses a service known as Aeronautical Telecommunications Network based on Open Systems Interconnection (ATN/OSI). The service is used to augment controller to pilot voice communications with rudimentary short text command and control messages. The service has seen successful deployment in a limited set of worldwide ATM domains.

The International Civil Aviation Organization (ICAO) is now undertaking the development of a next-generation replacement for ATN/OSI known as Aeronautical Telecommunications Network with Internet Protocol Services (ATN/IPS) [ATN][ATN-IPS]. ATN/IPS will eventually

provide an IPv6-based [RFC8200] service supporting pervasive ATM for Air Traffic Controllers (ATC), Airline Operations Controllers (AOC), and all commercial aircraft worldwide. As part of the ATN/IPS undertaking, a new mobile routing service will be needed. This document presents an approach based on the Border Gateway Protocol (BGP) [RFC4271].

Aircraft communicate via wireless aviation data links that typically support much lower data rates than terrestrial wireless and wired-line communications. For example, some Very High Frequency (VHF)-based data links only support data rates on the order of 32Kbps and an emerging L-Band data link that is expected to play a key role in future aeronautical communications only supports rates on the order of 1Mbps. Although satellite data links can provide much higher data rates during optimal conditions, like any other aviation data link they are subject to errors, delay, disruption, signal intermittence, degradation due to atmospheric conditions, etc. The well-connected ground domain ATN/IPS network should therefore treat each safety-of-flight critical packet produced by (or destined to) an aircraft as a precious commodity and strive for an optimized service that provides the highest possible degree of reliability. Furthermore, continuous performance-intensive control messaging services such as BGP peering sessions must be carried only over the well-connected ground domain ATN/IPS network and never over low-end aviation data links.

The ATN/IPS is an IP-based overlay network configured over one or more Internetworking underlays ("INETS") maintained by aeronautical network service providers such as ARINC, SITA and Inmarsat. The Overlay Multilink Network Interface (OMNI) [I-D.templin-6man-omni] uses an adaptation layer encapsulation to create a Non-Broadcast, Multiple Access (NBMA) virtual link spanning the entire ATN/IPS. Each aircraft connects to the OMNI link via an OMNI interface configured over the aircraft's underlying physical and/or virtual access network interfaces.

Each underlying INET comprises one or more "partitions" where all nodes within a partition can exchange packets with all other nodes, i.e., the partition is connected internally. There is no requirement that each INET partition uses the same IP protocol version nor has consistent IP addressing plans in comparison with other partitions. Instead, the OMNI link sees each partition as a "segment" of a link-layer topology concatenated by a service known as the OMNI Adaptation Layer (OAL) [I-D.templin-6man-omni] based on IPv6 encapsulation [RFC2473].

The IPv6 addressing architecture provides different classes of addresses, including Global Unicast Addresses (GUAs), Unique Local Addresses (ULAs) and Link-Local Addresses (LLAs) [RFC4291][RFC4193].

The ATN/IPS receives an IPv6 GUA Mobility Service Prefix (MSP) from an Internet assigned numbers authority, and each aircraft will receive a Mobile Network Prefix (MNP) delegation from the MSP that accompanies the aircraft wherever it travels. ATCs and AOCs will likewise receive MNPs, but they would typically appear in static (not mobile) deployments such as air traffic control towers, airline headquarters, etc. (Note that while IPv6 GUAs are assumed for ATN/IPS, IPv4 with public/private address could also be used.)

The adaptation layer uses ULAs in the source and destination addresses of adaptation layer IPv6 encapsulation headers. Each ULA includes an MNP in the interface identifier ("MNP-ULA"), as discussed in [I-D.templin-6man-omni]. Due to MNP delegation policies and random node mobility properties, MNP-ULAs are generally not aggregable in the BGP routing service and are represented as many more-specific prefixes instead of a smaller number of aggregated prefixes.

In addition, BGP routing service infrastructure nodes configure administratively-assigned ULAs ("ADM-ULA") that are statically-assigned and derived from a shorter ADM-ULA prefix assigned to their BGP network partitions. Unlike MNP-ULAs, the ADM-ULAs are persistently present and unchanging in the routing system. The BGP routing services therefore establish forwarding table entries based on these MNP-ULAs and ADM-ULAs instead of based on the GUA MNPs themselves. However, nodes set the 40-bit Global ID and 16-bit Subnet ID to 0 when they advertise MNP-ULAs in BGP routing exchanges and/or install MNP-ULAs in forwarding tables.

Both ADM-ULAs and MNP-ULAs are used by the OAL for nested encapsulation where the inner IPv6 packet is encapsulated in an IPv6 adaptation layer header with ULA source and destination addresses, which is then encapsulated in an IP header specific to the underlying Internetwork that will carry the actual packet transmission. A high level ATN/IPS network diagram is shown in Figure 1:

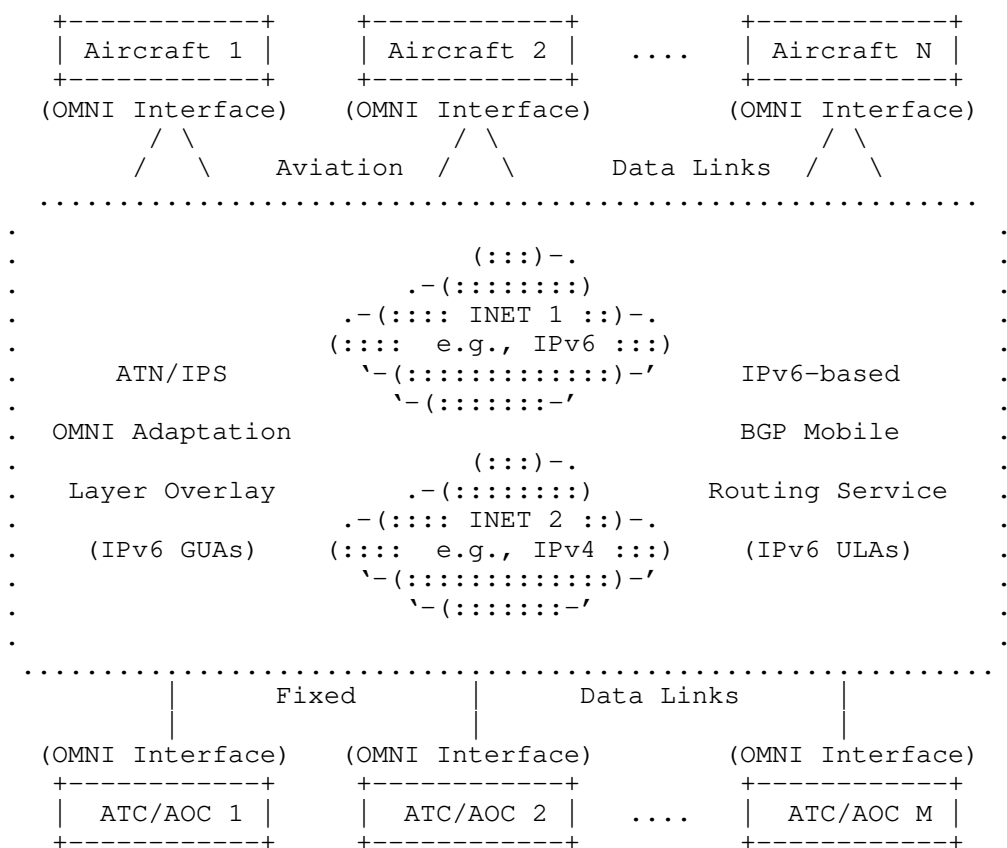


Figure 1: ATN/IPS Network Diagram

Connexion By Boeing [CBB] was an early aviation mobile routing service based on dynamic updates in the global public Internet BGP routing system. Practical experience with the approach has shown that frequent injections and withdrawals of prefixes in the Internet routing system can result in excessive BGP update messaging, slow routing table convergence times, and extended outages when no route is available. This is due to both conservative default BGP protocol timing parameters (see Section 6) and the complex peering interconnections of BGP routers within the global Internet infrastructure. The situation is further exacerbated by frequent aircraft mobility events that each result in BGP updates that must be propagated to all BGP routers in the Internet that carry a full routing table.

We therefore consider an approach using a BGP overlay network routing system where a private BGP routing protocol instance is maintained between ATN/IPS Autonomous System (AS) Border Routers (ASBRs). The private BGP instance does not interact with the native BGP routing systems in underlying INETs, and BGP updates are unidirectional from "stub" ASBRs (s-ASBRs) to a small set of "core" ASBRs (c-ASBRs) in a hub-and-spokes topology. No extensions to the BGP protocol are necessary, and BGP routing is based on (intermediate-layer) ULAs instead of upper- or lower-layer public/private IP prefixes. This allows ASBRs to perform adaptation layer forwarding based on intermediate layer IPv6 header information instead of network layer forwarding based on upper layer IP header information or link layer forwarding based on lower layer IP header information.

The s-ASBRs for each stub AS connect to a small number of c-ASBRs via dedicated high speed links and/or secured tunnels (e.g., IPsec [RFC4301], WireGuard [WG], etc.) over the underlying INET. Neighboring ASBRs should use also such IP layer security encapsulations over direct physical links to ensure INET layer security.

The s-ASBRs engage in external BGP (eBGP) peerings with their respective c-ASBRs, and only maintain routing table entries for the MNP-ULAs currently active within the stub AS. The s-ASBRs send BGP updates for MNP-ULA injections or withdrawals to c-ASBRs but do not receive any BGP updates from c-ASBRs. Instead, the s-ASBRs maintain default routes with their c-ASBRs as the next hop, and therefore hold only partial topology information.

The c-ASBRs connect to other c-ASBRs within the same partition using internal BGP (iBGP) peerings over which they collaboratively maintain a full routing table for all active MNP-ULAs currently in service within the partition. Therefore, only the c-ASBRs maintain a full BGP routing table and never send any BGP updates to s-ASBRs. This simple routing model therefore greatly reduces the number of BGP updates that need to be synchronized among peers, and the number is reduced further still when intradomain routing changes within stub ASes are processed within the AS instead of being propagated to the core. BGP Route Reflectors (RRs) [RFC4456] can also be used to support increased scaling properties.

When there are multiple INET partitions, the c-ASBRs of each partition use eBGP to peer with the c-ASBRs of other partitions so that the full set of ULAs for all partitions are known globally among all of the c-ASBRs. Each c/s-ASBR further configures an ADM-ULA which is taken from an ADM-ULA prefix assigned to each partition, as well as static forwarding table entries for all other OMNI link partition prefixes. Both ADM-ULAs and MNP-ULAs are used by the OAL

for nested encapsulation where the inner IPv6 packet is encapsulated in an IPv6 OAL header with ULA source and destination addresses, which is then encapsulated in an IP header specific to the INET partition.

With these intra- and inter-INET BGP peerings in place, a forwarding plane spanning tree is established that properly covers the entire operating domain. All nodes in the network can be visited using strict spanning tree hops, but in many instances this may result in longer paths than are necessary. AERO [I-D.templin-6man-aero] provides an example service for discovering and utilizing (route-optimized) shortcuts that do not always follow strict spanning tree paths.

The remainder of this document discusses the proposed BGP-based ATN/IPS mobile routing service.

2. Terminology

The terms Autonomous System (AS) and Autonomous System Border Router (ASBR) are the same as defined in [RFC4271].

The following terms are defined for the purposes of this document:

Air Traffic Management (ATM)

The worldwide service for coordinating safe aviation operations.

Air Traffic Controller (ATC)

A government agent responsible for coordinating with aircraft within a defined operational region via voice and/or data Command and Control messaging.

Airline Operations Controller (AOC)

An airline agent responsible for tracking and coordinating with aircraft within their fleet.

Aeronautical Telecommunications Network with Internet Protocol Services (ATN/IPS)

A future aviation network for ATCs and AOCs to coordinate with all aircraft operating worldwide. The ATN/IPS will be an IPv6-based overlay network service that connects access networks via tunneling over one or more Internetworking underlays.

Internetworking underlay ("INET")

A wide-area network that supports overlay network tunneling and connects Radio Access Networks to the rest of the ATN/IPS. Example INET service providers for civil aviation include ARINC, SITA and Inmarsat.

(Radio) Access Network ("ANET")

An aviation radio data link service provider's network, including radio transmitters and receivers as well as supporting ground-domain infrastructure needed to convey a customer's data packets to outside INETs. The term ANET is intended in the same spirit as for radio-based Internet service provider networks (e.g., cellular operators), but can also refer to ground-domain networks that connect AOCs and ATCs.

partition (or "segment")

A fully-connected internal subnetwork of an INET in which all nodes can communicate with all other nodes within the same partition using the same IP protocol version and addressing plan. Each INET consists of one or more partitions.

Overlay Multilink Network Interface (OMNI)

A virtual layer 2 bridging service that presents an ATN/IPS overlay unified link view even though the underlay may consist of multiple INET partitions. The OMNI virtual link is manifested through nested encapsulation in which original IP packets from the ATN/IPS are first encapsulated in ULA-addressed IPv6 headers which are then forwarded to the next hop using INET encapsulation if necessary. Forwarding over the OMNI virtual link is therefore based on ULAs instead of the original IP addresses. In this way, packets sent from a source can be conveyed over the OMNI virtual link even though there may be many underlying INET partitions in the path to the destination.

OMNI Adaptation Layer (OAL)

A middle layer below the IP layer but above the INET layer that applies IP-in-IPv6 encapsulation prior to INET encapsulation. The IPv6 encapsulation header inserted by the OAL uses ULAs instead of GUAs. End systems that configure OMNI interfaces act as OAL ingress and egress points, while intermediate systems with OMNI interfaces act as OAL forwarding nodes. There may be zero, one or many intermediate nodes between the OAL ingress and egress, but the upper layer IPv6 Hop Limit is not decremented during (OAL layer) forwarding. Further details on OMNI and the OAL are found in [I-D.templin-6man-omni].

OAL Autonomous System (OAL AS)

A "hub-of-hubs" autonomous system maintained through peerings between the core autonomous systems of different OMNI virtual link partitions.

Core Autonomous System Border Router (c-ASBR)

A BGP router located in the hub of the INET partition hub-and-spokes overlay network topology.

Core Autonomous System (Core AS)

The "hub" autonomous system maintained by all c-ASBRs within the same partition.

Stub Autonomous System Border Router (s-ASBR)

A BGP router configured as a spoke in the INET partition hub-and-spokes overlay network topology.

Stub Autonomous System (Stub AS)

A logical grouping that includes all Clients currently associated with a given s-ASBR.

Client

An ATC, AOC or aircraft that connects to the ATN/IPS as a leaf node. The Client could be a singleton host, or a router that connects a mobile or fixed network.

Proxy/Server

An ANET/INET border node that acts as a transparent intermediary between Clients and s-ASBRs. From the Client's perspective, the Proxy/Server presents the appearance that the Client is communicating directly with the s-ASBR. From the s-ASBR's perspective, the Proxy/Server presents the appearance that the s-ASBR is communicating directly with the Client.

Mobile Network Prefix (MNP)

An IPv6 prefix that is delegated to any ATN/IPS end system, including ATCs, AOCs, and aircraft.

Mobility Service Prefix (MSP)

An aggregated IP prefix assigned to the ATN/IPS by an Internet assigned numbers authority, and from which all MNPs are delegated (e.g., up to 2^{32} IPv6 /56 MNPs could be delegated from a /24 MSP).

3. ATN/IPS Routing System

The ATN/IPS routing system comprises a private BGP instance coordinated in an overlay network via tunnels between neighboring ASBRs over one or more underlying INETs. The ATN/IPS routing system interacts with underlying INET BGP routing systems only through the static advertisement of a small and unchanging set of MSPs instead of the full dynamically changing set of MNPs.

Within each INET partition, each s-ASBR connects a stub AS to the INET partition core using a distinct stub AS Number (ASN). Each s-ASBR further uses eBGP to peer with one or more c-ASBRs. All c-ASBRs are members of the INET partition core AS, and use a shared

core ASN. Unique ASNs are assigned according to the standard 32-bit ASN format [RFC4271][RFC6793]. Since the BGP instance does not connect with any INET BGP routing systems, the ASNs can be assigned from the [RFC6996] 32-bit ASN space which reserves 94,967,295 numbers for private use. The ASNs must be allocated and managed by an ATN/IPS assigned numbers authority established by ICAO, which must ensure that ASNs are responsibly distributed without duplication and/or overlap.

The c-ASBRs use iBGP to maintain a synchronized consistent view of all active MNP-ULAs currently in service within the INET partition. Figure 2 below represents the reference INET partition deployment. (Note that the figure shows details for only two s-ASBRs (s-ASBR1 and s-ASBR2) due to space constraints, but the other s-ASBRs should be understood to have similar Stub AS, MNP and eBGP peering arrangements.) The solution described in this document is flexible enough to extend to these topologies.

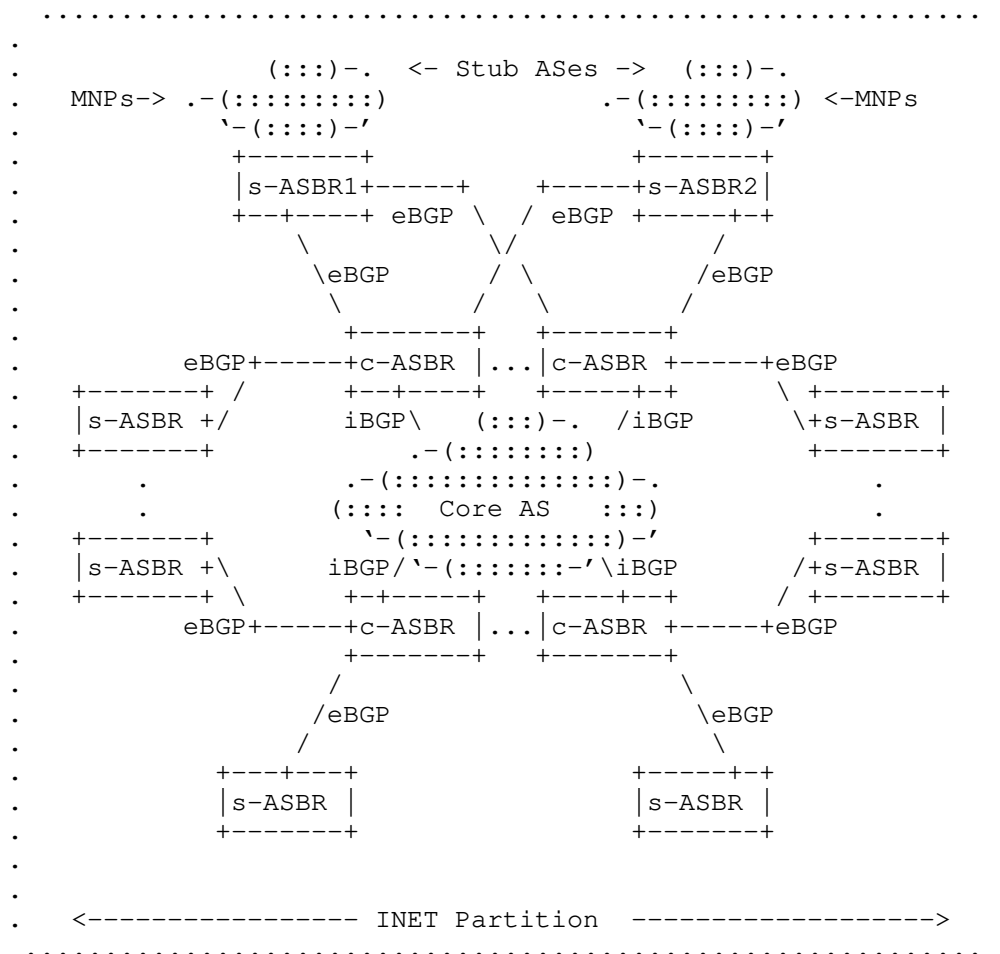


Figure 2: INET Partition Reference Deployment

In the reference deployment, each s-ASBR maintains routes for active MNP-ULAs that currently belong to its stub AS. In response to "Inter-domain" mobility events, each s-ASBR dynamically announces new MNP-ULAs and withdraws departed MNP-ULAs in its eBGP updates to c-ASBRs. Since ATN/IPS end systems are expected to remain within the same stub AS for extended timeframes, however, intra-domain mobility events (such as an aircraft handing off between cell towers) are handled within the stub AS instead of being propagated as inter-domain eBGP updates.

Each c-ASBR configures a black-hole route for each of its MSPs. By black-holing the MSPs, the c-ASBR maintains forwarding table entries only for the MNP-ULAs that are currently active. If an arriving packet matches a black-hole route without matching an MNP-ULA, the c-ASBR should drop the packet and may also generate an ICMPv6 Destination Unreachable message [RFC4443], i.e., without forwarding the packet outside of the ATN/IPS overlay based on a less-specific route.

The c-ASBRs do not send BGP updates for MNP-ULAs to s-ASBRs, but instead originate a default route. In this way, s-ASBRs have only partial topology knowledge (i.e., they know only about the active MNP-ULAs currently within their stub ASes) and they forward all other packets to c-ASBRs which have full topology knowledge.

Each s-ASBR and c-ASBR configures an ADM-ULA that is aggregable within an INET partition, and each partition configures a unique ADM-ULA prefix that is permanently announced into the routing system. The core ASes of each INET partition are joined together through external BGP peerings. The c-ASBRs of each partition establish external peerings with the c-ASBRs of other partitions to form a "core-of-cores" OMNI link AS. The OMNI link AS contains the global knowledge of all MNP-ULAs deployed worldwide, and supports ATN/IPS overlay communications between nodes located in different INET partitions by virtue of OAL encapsulation. OMNI link nodes can then navigate to ASBRs by including an ADM-ULA or directly to an end system by including an MNP-ULA in the destination address of an OAL-encapsulated packet (see: [I-D.templin-6man-aero]). Figure 3 shows a reference OAL topology.

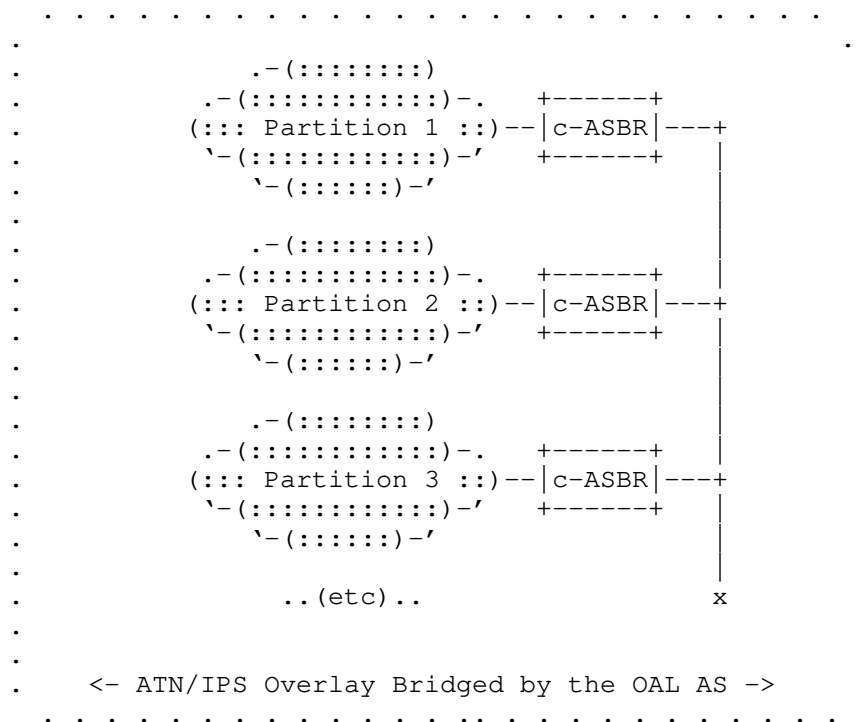


Figure 3: Spanning Partitions with the OAL

Scaling properties of this ATN/IPS routing system are limited by the number of BGP routes that can be carried by the c-ASBRs. A 2015 study showed that BGP routers in the global public Internet at that time carried more than 500K routes with linear growth and no signs of router resource exhaustion [BGP]. A more recent network emulation study also showed that a single c-ASBR can accommodate at least 1M dynamically changing BGP routes even on a lightweight virtual machine. Commercially-available high-performance dedicated router hardware can support many millions of routes.

Therefore, assuming each c-ASBR can carry 1M or more routes, this means that at least 1M ATN/IPS end system MNP-ULAs can be serviced by a single set of c-ASBRs and that number could be further increased by using RRs and/or more powerful routers. Another means of increasing scale would be to assign a different set of c-ASBRs for each set of MSPs. In that case, each s-ASBR still peers with one or more c-ASBRs from each set of c-ASBRs, but the s-ASBR institutes route filters so that it only sends BGP updates to the specific set of c-ASBRs that aggregate the MSP. In this way, each set of c-ASBRs maintains separate routing and forwarding tables so that scaling is distributed

across multiple c-ASBR sets instead of concentrated in a single c-ASBR set. For example, a first c-ASBR set could aggregate an MSP segment A::/32, a second set could aggregate B::/32, a third could aggregate C::/32, etc. The union of all MSP segments would then constitute the collective MSP(s) for the entire ATN/IPS, with potential for supporting many millions of mobile networks or more.

In this way, each set of c-ASBRs services a specific set of MSPs, and each s-ASBR configures MSP-specific routes that list the correct set of c-ASBRs as next hops. This design also allows for natural incremental deployment, and can support initial medium-scale deployments followed by dynamic deployment of additional ATN/IPS infrastructure elements without disturbing the already-deployed base. For example, a few more c-ASBRs could be added if the MNP service demand ever outgrows the initial deployment. For larger-scale applications (such as unmanned air vehicles and terrestrial vehicles) even larger scales can be accommodated by adding more c-ASBRs.

4. ATN/IPS (Radio) Access Network (ANET) Model

(Radio) Access Networks (ANETs) connect end system Clients such as aircraft, ATCs, AOCs etc. to the ATN/IPS routing system. Clients may connect to multiple ANETs at once, for example, when they have both satellite and cellular data links activated simultaneously. Clients configure an Overlay Multilink Network (OMNI) Interface [I-D.templin-6man-omni] over their underlying ANET interfaces as a connection to an NBMA virtual link (manifested by the OAL) that spans the entire ATN/IPS. Clients may further move between ANETs in a manner that is perceived as a network layer mobility event. Clients could therefore employ a multilink/mobility routing service such as those discussed in Section 7.

Clients register all of their active data link connections with their serving s-ASBRs as discussed in Section 3. Clients may connect to s-ASBRs either directly, or via a Proxy/Server at the ANET/INET boundary.

Figure 4 shows the ATN/IPS ANET model where Clients connect to ANETs via aviation data links. Clients register their ANET addresses with a nearby s-ASBR, where the registration process may be brokered by a Proxy/Server at the edge of the ANET.

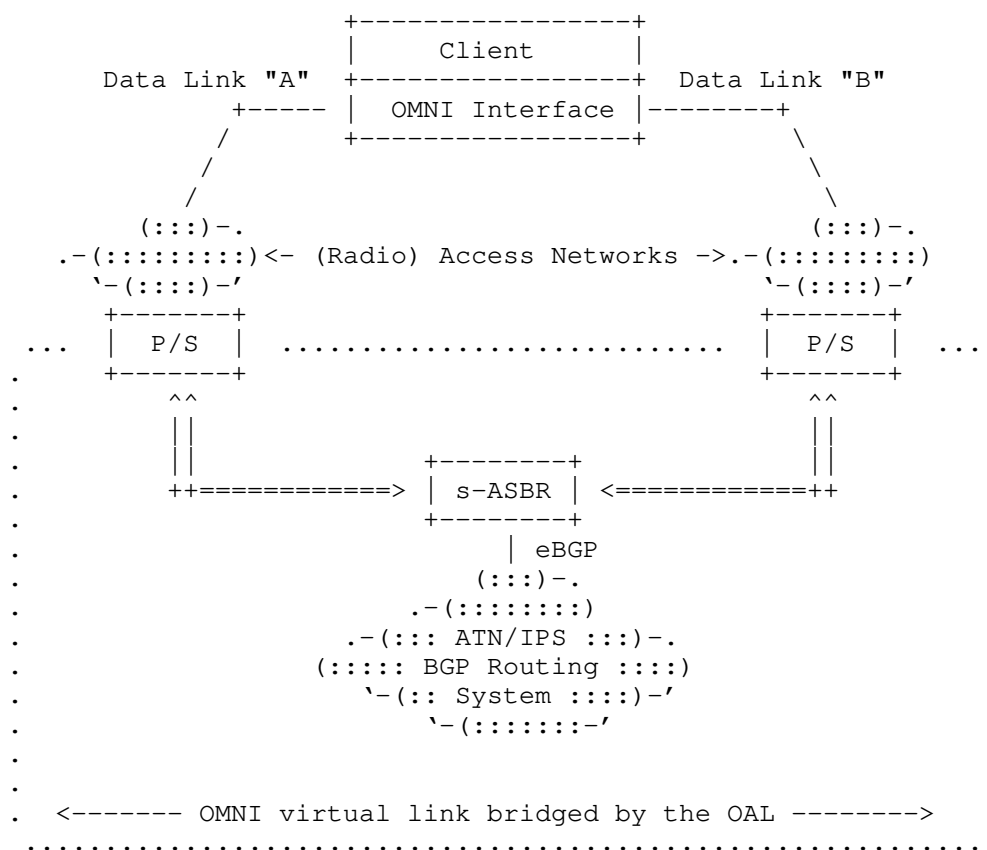


Figure 4: ATN/IPS ANET Architecture

When a Client connects to an ANET it specifies a nearby s-ASBR that it has selected to connect to the ATN/IPS. The login process is transparently brokered by a Proxy/Server at the border of the ANET which then conveys the connection request to the s-ASBR via tunneling across the OMNI virtual link. Each ANET border Proxy/Server is also equally capable of serving in the s-ASBR role so that a first on-link Proxy/Server can be selected as the s-ASBR while all others perform the Proxy/Server role in a hub-and-spokes arrangement. An on-link Proxy/Server is selected to serve the s-ASBR role when it receives a control message from a Client requesting that service.

The Client can coordinate with a network-based s-ASBR over additional ANETs after it has already coordinated with a first-hop Proxy/Server over a first ANET. If the Client connects to multiple ANETs, the s-ASBR will register the individual ANET Proxy/Servers as conduits through which the Client can be reached. The Client then sees the

s-ASBR as the "hub" in a "hub-and-spokes" arrangement with the first-hop Proxy/Servers as spokes. Selection of a network-based s-ASBR is through the discovery methods specified in relevant mobility and virtual link coordination specifications (e.g., see AERO [I-D.templin-6man-aero] and OMNI [I-D.templin-6man-omni]).

The s-ASBR represents all of its active Clients as MNP-ULA routes in the ATN/IPS BGP routing system. The s-ASBR's stub AS is therefore used only to advertise the set of MNPs of all its active Clients to its BGP peer c-ASBRs and not to peer with other s-ASBRs (i.e., the stub AS is a logical construct and not a physical one). The s-ASBR injects the MNP-ULAs of its active Clients and withdraws the MNP-ULAs of its departed Clients via BGP updates to c-ASBRs, which further propagate the MNP-ULAs to other c-ASBRs within the OAL AS. Since Clients are expected to remain associated with their current s-ASBR for extended periods, the level of MNP-ULA injections and withdrawals in the BGP routing system will be on the order of the numbers of network joins, leaves and s-ASBR handovers for aircraft operations (see: Section 6). It is important to observe that fine-grained events such as Client mobility and Quality of Service (QoS) signaling are coordinated only by Proxies and the Client's current s-ASBRs, and do not involve other ASBRs in the routing system. In this way, intradomain routing changes within the stub AS are not propagated into the rest of the ATN/IPS BGP routing system.

5. ATN/IPS Route Optimization

ATN/IPS end systems will frequently need to communicate with correspondents associated with other s-ASBRs. In the BGP peering topology discussed in Section 3, this can initially only be accommodated by including multiple extraneous hops and/or spanning tree segments in the forwarding path. In many cases, it would be desirable to establish a "short cut" around this "dogleg" route so that packets can traverse a minimum number of tunneling hops across the OMNI virtual link. ATN/IPS end systems could therefore employ a route optimization service according to the mobility service employed (see: Section 7).

Each s-ASBR provides designated routing services for only a subset of all active Clients, and instead acts as a simple Proxy/Server for other Clients. As a designated router, the s-ASBR advertises the MNPs of each of its active Clients into the ATN/IPS routing system and provides basic (unoptimized) forwarding services when necessary. An s-ASBR could be the first-hop ATN/IPS service access point for some, all or none of a Client's underlying interfaces, while the Client's other underlying interfaces employ the Proxy/Server function of other s-ASBRs. Route optimization allows Client-to-Client communications while bypassing s-ASBR designated routing services whenever possible.

A route optimization example is shown in Figure 5 and Figure 6 below. In the first figure, multiple spanning tree segments between Proxy/Servers and ASBRs are necessary to convey packets between Clients associated with different s-ASBRs. In the second figure, the optimized route tunnels packets directly between Proxy/Servers without involving the ASBRs.

These route optimized paths are established through secured control plane messaging (i.e., over secured tunnels and/or using higher-layer control message authentications) but do not provide lower-layer security for the data plane. Data communications over these route optimized paths should therefore employ higher-layer security.

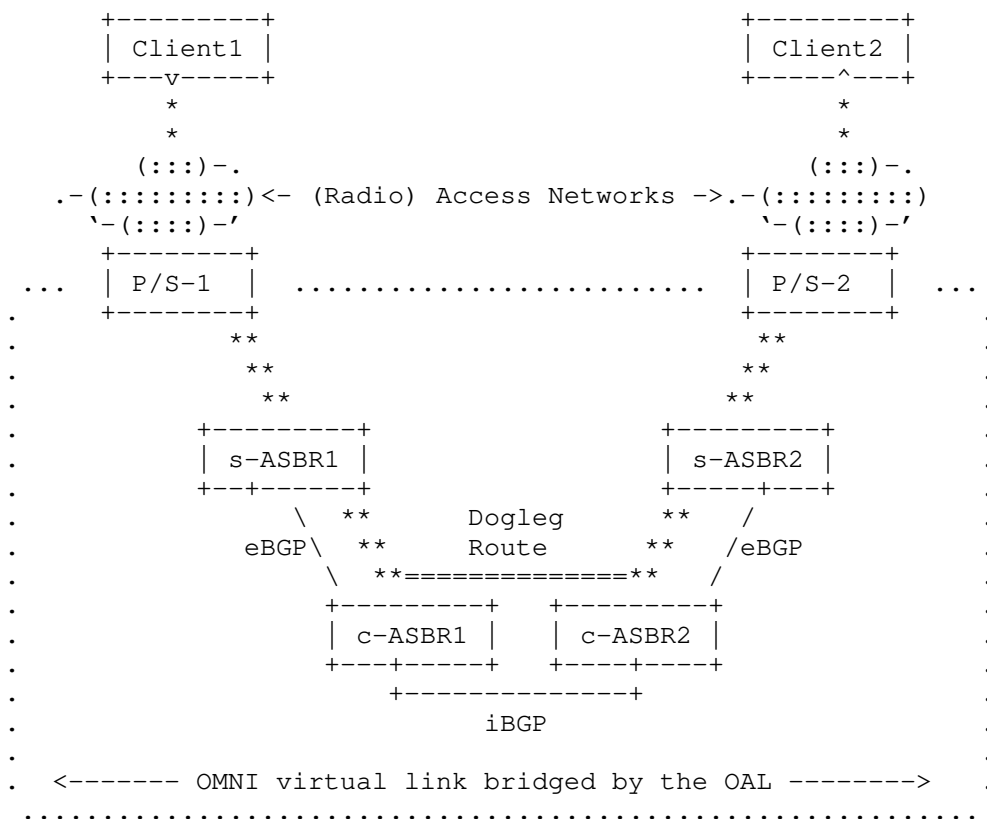


Figure 5: Dogleg Route Before Optimization

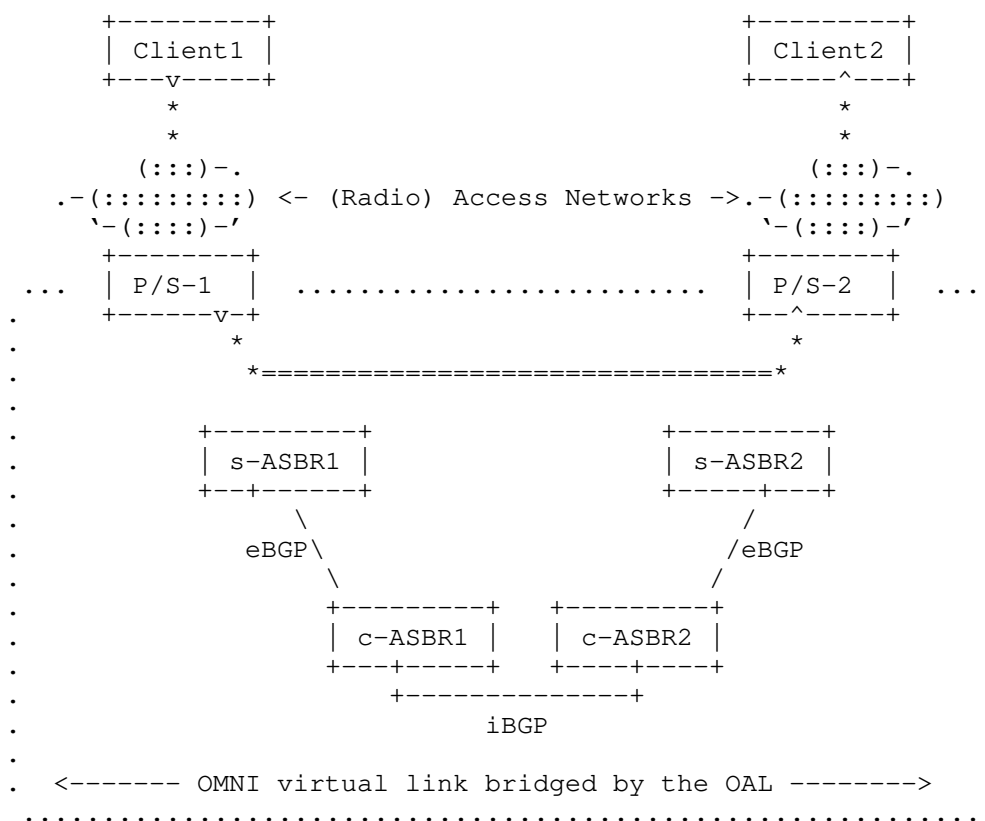


Figure 6: Optimized Route

6. BGP Protocol Considerations

The number of eBGP peering sessions that each c-ASBR must service is proportional to the number of s-ASBRs in its local partition. Network emulations with lightweight virtual machines have shown that a single c-ASBR can service at least 100 eBGP peerings from s-ASBRs that each advertise 10K MNP-ULA routes (i.e., 1M total). It is expected that robust c-ASBRs can service many more peerings than this - possibly by multiple orders of magnitude. But even assuming a conservative limit, the number of s-ASBRs could be increased by also increasing the number of c-ASBRs. Since c-ASBRs also peer with each other using iBGP, however, larger-scale c-ASBR deployments may need to employ an adjunct facility such as BGP Route Reflectors (RRs) [RFC4456].

The number of aircraft in operation at a given time worldwide is likely to be significantly less than 1M, but we will assume this number for a worst-case analysis. Assuming a worst-case average 1 hour flight profile from gate-to-gate with 10 service region transitions per flight, the entire system will need to service at most 10M BGP updates per hour (2778 updates per second). This number is within the realm of the peak BGP update messaging seen in the global public Internet today [BGP2]. Assuming a BGP update message size of 100 bytes (800bits), the total amount of BGP control message traffic to a single c-ASBR will be less than 2.5Mbps which is a nominal rate for modern data links.

Industry standard BGP routers provide configurable parameters with conservative default values. For example, the default hold time is 90 seconds, the default keepalive time is 1/3 of the hold time, and the default MinRouteAdvertisementInterval is 30 seconds for eBGP peers and 5 seconds for iBGP peers (see Section 10 of [RFC4271]). For the simple mobile routing system described herein, these parameters can be set to more aggressive values to support faster neighbor/link failure detection and faster routing protocol convergence times. For example, a hold time of 3 seconds and a MinRouteAdvertisementInterval of 0 seconds for both iBGP and eBGP.

Instead of adjusting BGP default time values, BGP routers can use the Bidirectional Forwarding Detection (BFD) protocol [RFC5880] to quickly detect link failures that don't result in interface state changes, BGP peer failures, and administrative state changes. BFD is important in environments where rapid response to failures is required for routing reconvergence and, hence, communications continuity.

Each c-ASBR will be using eBGP both in the ATN/IPS and the INET with the ATN/IPS unicast IPv6 routes resolving over INET routes. Consequently, c-ASBRs and potentially s-ASBRs will need to support separate local ASes for the two BGP routing domains and routing policy or assure routes are not propagated between the two BGP routing domains. From a conceptual, operational and correctness standpoint, the implementation should provide isolation between the two BGP routing domains (e.g., separate BGP instances).

ADM-ULAs and MNP-ULAs begin with fd00::/8 followed by a pseudo-random 40-bit global ID to form the prefix [ULA]::/48, along with a 16-bit Subnet ID '*' to form the prefix [ULA*]::/64. Each individual address taken from [ULA*]::/64 includes additional routing information in the interface identifier. For example, for the MNP 2001:db8:1:0::/56, the resulting MNP-ULA is [ULA*]:2001:db8:1:0/120, and for the administrative address 1001:2002 the ADM-ULA is [ULA*]:1001:2002/64 (see: [I-D.templin-6man-omni] for further

details). However, MNP-ULA prefixes installed in the BGP routing system always set the Global ID and Subnet ID to 0 (i.e., the "wildcard" subnet) since OMNI link forwarding decisions are based solely on the MNP found in the interface identifier independently of the Global/Subnet IDs.

This gives rise to a BGP routing system that must accommodate large numbers of long and non-aggregable MNP-ULA prefixes as well as moderate numbers of long and semi-aggregable ADM-ULA prefixes. The system is kept stable and scalable through the s-ASBR / c-ASBR hub-and-spokes topology which ensures that mobility-related churn is not exposed to the core.

7. Stub AS Mobile Routing Services

Stub ASes maintain intradomain routing information for mobile node clients, and are responsible for all localized mobility signaling without disturbing the BGP routing system. Clients can enlist the services of a candidate mobility service such as Mobile IPv6 (MIPv6) [RFC6275], LISP [I-D.ietf-lisp-rfc6830bis] or AERO [I-D.templin-6man-aero] according to the service offered by the stub AS. Further details of mobile routing services are out of scope for this document.

8. Implementation Status

The BGP routing topology described in this document has been modeled in realistic network emulations showing that at least 1 million MNP-ULAs can be propagated to each c-ASBR even on lightweight virtual machines. No BGP routing protocol extensions need to be adopted.

9. IANA Considerations

This document does not introduce any IANA considerations.

10. Security Considerations

ATN/IPS ASBRs on the open Internet are susceptible to the same attack profiles as for any Internet nodes. For this reason, ASBRs should employ physical security and/or IP securing mechanisms such as IPsec [RFC4301], WireGuard [WG], etc.

ATN/IPS ASBRs present targets for Distributed Denial of Service (DDoS) attacks. This concern is no different than for any node on the open Internet, where attackers could send spoofed packets to the node at high data rates. This can be mitigated by connecting ATN/IPS ASBRs over dedicated links with no connections to the Internet and/or when ASBR connections to the Internet are only permitted through well-managed firewalls.

ATN/IPS s-ASBRs should institute rate limits to protect low data rate aviation data links from receiving DDoS packet floods.

BGP protocol message exchanges and control message exchanges used for route optimization must be secured to ensure the integrity of the system-wide routing information base. Security is based on IP layer security associations between peers which ensure confidentiality, integrity and authentication over secured tunnels (see above). Higher layer security protection such as TCP-AO [RFC5926] is therefore optional, since it would be redundant with the security provided at lower layers.

Data communications over route optimized paths should employ end-to-end higher-layer security since only the control plane and unoptimized paths are protected by lower-layer security. End-to-end higher-layer security mechanisms include QUIC-TLS [RFC9001], TLS [RFC8446], DTLS [RFC6347], SSH [RFC4251], etc. applied in a manner outside the scope of this document.

This document does not include any new specific requirements for mitigation of DDoS.

10.1. Public Key Infrastructure (PKI) Considerations

In development of the overall ATN/IPS operational concept, ICAO addressed the security concerns in multiple ways to ensure coordination and consistency across the various groups. This also avoided potential duplicative work. Technical provisions related specifically to the operation of ATN/IPS are specified in supporting ATN/IPS standards. However, other considerations such as the establishment of a PKI, were determined to have an impact beyond ATN/IPS. ICAO created a Trust Framework Study Group (TFSG) to define various governance, policy, procedures and overall technical performance requirements for system connectivity and interoperability.

As part of their charter, the TSFG is specifically developing a concept of operations for a common aviation digital trust framework and principles to facilitate an interoperable secure, cyber resilient and seamless exchange of information in a digitally connected

environment. They are also developing governance principles, policy, procedures and requirements for establishing digital identity for a global trust framework that will consider any exchange of information among users of the aviation ecosystem, and to promote these concepts with all relevant stakeholders.

ATN/IPS will take advantage of the developments of TFSG within the overall ATN/IPS operational concept. As such, this will include the usage of the PKI specification resulting from the TFSG.

11. Acknowledgements

This work is aligned with the FAA as per the SE2025 contract number DTFAWA-15-D-00030.

This work is aligned with the NASA Safe Autonomous Systems Operation (SASO) program under NASA contract number NNA16BD84C.

This work is aligned with the Boeing Commercial Airplanes (BCA) Internet of Things (IoT) and autonomy programs.

This work is aligned with the Boeing Information Technology (BIT) MobileNet program.

The following individuals contributed insights that have improved the document: Ahmad Amin, Mach Chen, Russ Housley, Erik Kline, Hubert Kuenig, Tony Li, Gyan Mishra, Alexandre Petrescu, Dave Thaler, Pascal Thubert, Michael Tuxen, Tony Whyman.

12. References

12.1. Normative References

- [RFC2473] Conta, A. and S. Deering, "Generic Packet Tunneling in IPv6 Specification", RFC 2473, DOI 10.17487/RFC2473, December 1998, <<https://www.rfc-editor.org/info/rfc2473>>.
- [RFC4193] Hinden, R. and B. Haberman, "Unique Local IPv6 Unicast Addresses", RFC 4193, DOI 10.17487/RFC4193, October 2005, <<https://www.rfc-editor.org/info/rfc4193>>.
- [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>>.

- [RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, DOI 10.17487/RFC4291, February 2006, <<https://www.rfc-editor.org/info/rfc4291>>.
- [RFC4443] Conta, A., Deering, S., and M. Gupta, Ed., "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", STD 89, RFC 4443, DOI 10.17487/RFC4443, March 2006, <<https://www.rfc-editor.org/info/rfc4443>>.
- [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>>.
- [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>>.
- [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>>.

12.2. Informative References

- [ATN] Maiolla, V., "The OMNI Interface - An IPv6 Air/Ground Interface for Civil Aviation, IETF Liaison Statement #1676, <https://datatracker.ietf.org/liaison/1676/>", 3 March 2020.
- [ATN-IPS] WG-I, ICAO., "ICAO Document 9896 (Manual on the Aeronautical Telecommunication Network (ATN) using Internet Protocol Suite (IPS) Standards and Protocol), Draft Edition 3 (work-in-progress)", 10 December 2020.
- [BGP] Huston, G., "BGP in 2015, <http://potaroo.net>", January 2016.
- [BGP2] Huston, G., "BGP Instability Report, <http://bgpupdates.potaroo.net/instability/bgpupd.html>", May 2017.
- [CBB] Dul, A., "Global IP Network Mobility using Border Gateway Protocol (BGP), http://www.quark.net/docs/Global_IP_Network_Mobility_using_BGP.pdf", March 2006.

[I-D.ietf-lisp-rfc6830bis]

Farinacci, D., Fuller, V., Meyer, D., Lewis, D., and A. Cabellos, "The Locator/ID Separation Protocol (LISP)", Work in Progress, Internet-Draft, draft-ietf-lisp-rfc6830bis-36, 18 November 2020, <<https://www.ietf.org/archive/id/draft-ietf-lisp-rfc6830bis-36.txt>>.

[I-D.templin-6man-aero]

Templin, F. L., "Automatic Extended Route Optimization (AERO)", Work in Progress, Internet-Draft, draft-templin-6man-aero-42, 9 April 2022, <<https://www.ietf.org/archive/id/draft-templin-6man-aero-42.txt>>.

[I-D.templin-6man-omni]

Templin, F. L., "Transmission of IP Packets over Overlay Multilink Network (OMNI) Interfaces", Work in Progress, Internet-Draft, draft-templin-6man-omni-57, 9 April 2022, <<https://www.ietf.org/archive/id/draft-templin-6man-omni-57.txt>>.

[RFC2784] Farinacci, D., Li, T., Hanks, S., Meyer, D., and P. Traina, "Generic Routing Encapsulation (GRE)", RFC 2784, DOI 10.17487/RFC2784, March 2000, <<https://www.rfc-editor.org/info/rfc2784>>.

[RFC4251] Ylonen, T. and C. Lonvick, Ed., "The Secure Shell (SSH) Protocol Architecture", RFC 4251, DOI 10.17487/RFC4251, January 2006, <<https://www.rfc-editor.org/info/rfc4251>>.

[RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, DOI 10.17487/RFC4301, December 2005, <<https://www.rfc-editor.org/info/rfc4301>>.

[RFC5926] Lebovitz, G. and E. Rescorla, "Cryptographic Algorithms for the TCP Authentication Option (TCP-AO)", RFC 5926, DOI 10.17487/RFC5926, June 2010, <<https://www.rfc-editor.org/info/rfc5926>>.

[RFC6275] Perkins, C., Ed., Johnson, D., and J. Arkko, "Mobility Support in IPv6", RFC 6275, DOI 10.17487/RFC6275, July 2011, <<https://www.rfc-editor.org/info/rfc6275>>.

[RFC6347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security Version 1.2", RFC 6347, DOI 10.17487/RFC6347, January 2012, <<https://www.rfc-editor.org/info/rfc6347>>.

- [RFC6793] Vohra, Q. and E. Chen, "BGP Support for Four-Octet Autonomous System (AS) Number Space", RFC 6793, DOI 10.17487/RFC6793, December 2012, <<https://www.rfc-editor.org/info/rfc6793>>.
- [RFC6996] Mitchell, J., "Autonomous System (AS) Reservation for Private Use", BCP 6, RFC 6996, DOI 10.17487/RFC6996, July 2013, <<https://www.rfc-editor.org/info/rfc6996>>.
- [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>>.
- [RFC9001] Thomson, M., Ed. and S. Turner, Ed., "Using TLS to Secure QUIC", RFC 9001, DOI 10.17487/RFC9001, May 2021, <<https://www.rfc-editor.org/info/rfc9001>>.
- [WG] Donenfeld, J., "WireGuard: Fast, Modern, Secure VPN Tunnel, <https://www.wireguard.com/>", February 2022.

Appendix A. BGP Convergence Considerations

Experimental evidence has shown that BGP convergence time required after an MNP-ULA is asserted at a new location or withdrawn from an old location can be several hundred milliseconds even under optimal AS peering arrangements. This means that packets in flight destined to an MNP-ULA route that has recently been changed can be (mis)delivered to an old s-ASBR after a Client has moved to a new s-ASBR.

To address this issue, the old s-ASBR can maintain temporary state for a "departed" Client that includes an OAL address for the new s-ASBR. The OAL address never changes since ASBRs are fixed infrastructure elements that never move. Hence, packets arriving at the old s-ASBR can be forwarded to the new s-ASBR while the BGP routing system is still undergoing reconvergence. Therefore, as long as the Client associates with the new s-ASBR before it departs from the old s-ASBR (while informing the old s-ASBR of its new location) packets in flight during the BGP reconvergence window are accommodated without loss.

Appendix B. Change Log

<< RFC Editor - remove prior to publication >>

Differences from earlier versions:

* Submit for RFC publication.

Authors' Addresses

Fred L. Templin (editor)
Boeing Research & Technology
P.O. Box 3707
Seattle, WA 98124
United States of America
Email: fltemplin@acm.org

Greg Saccone
Boeing Research & Technology
P.O. Box 3707
Seattle, WA 98124
United States of America
Email: gregory.t.saccone@boeing.com

Gaurav Dawra
LinkedIn
United States of America
Email: gdawra.ietf@gmail.com

Acee Lindem
Cisco Systems, Inc.
United States of America
Email: acee@cisco.com

Victor Moreno
Cisco Systems, Inc.
United States of America
Email: vimoreno@cisco.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.

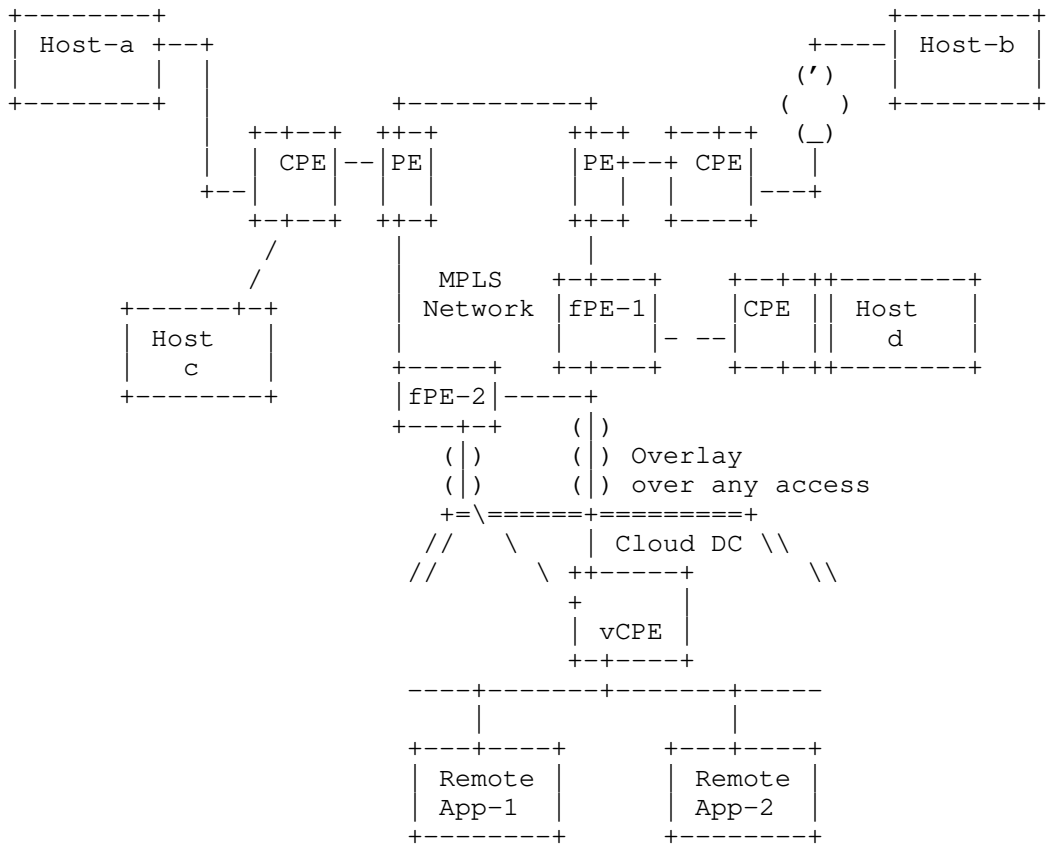


Figure 1: VPN Extension to Cloud DC

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.

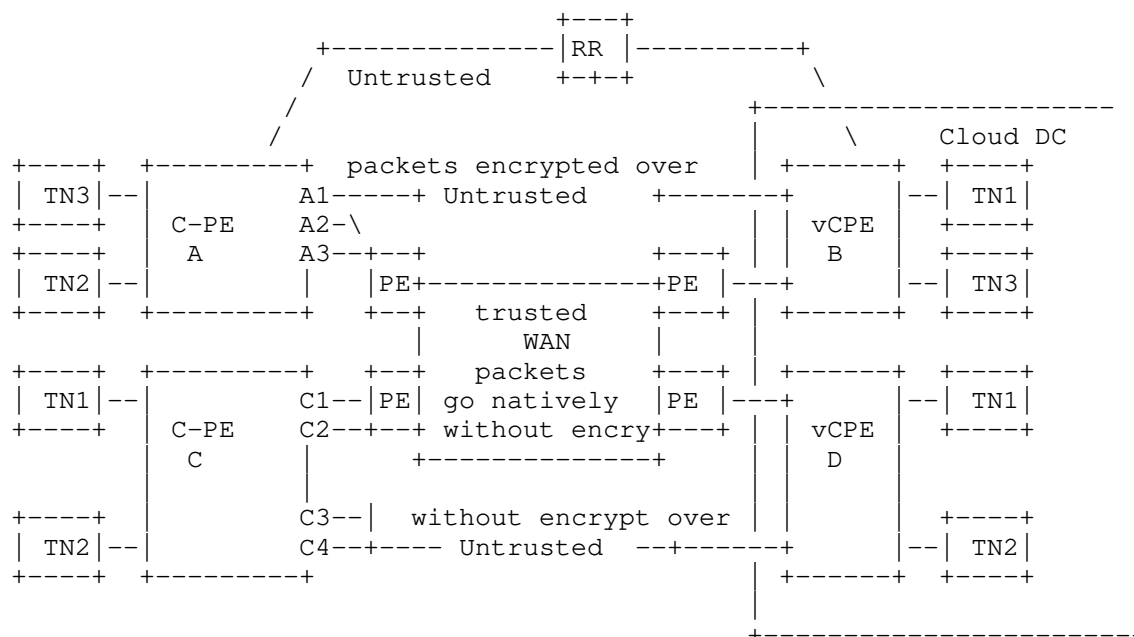


Figure 2: vCPEs reached by Hybrid Paths

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.

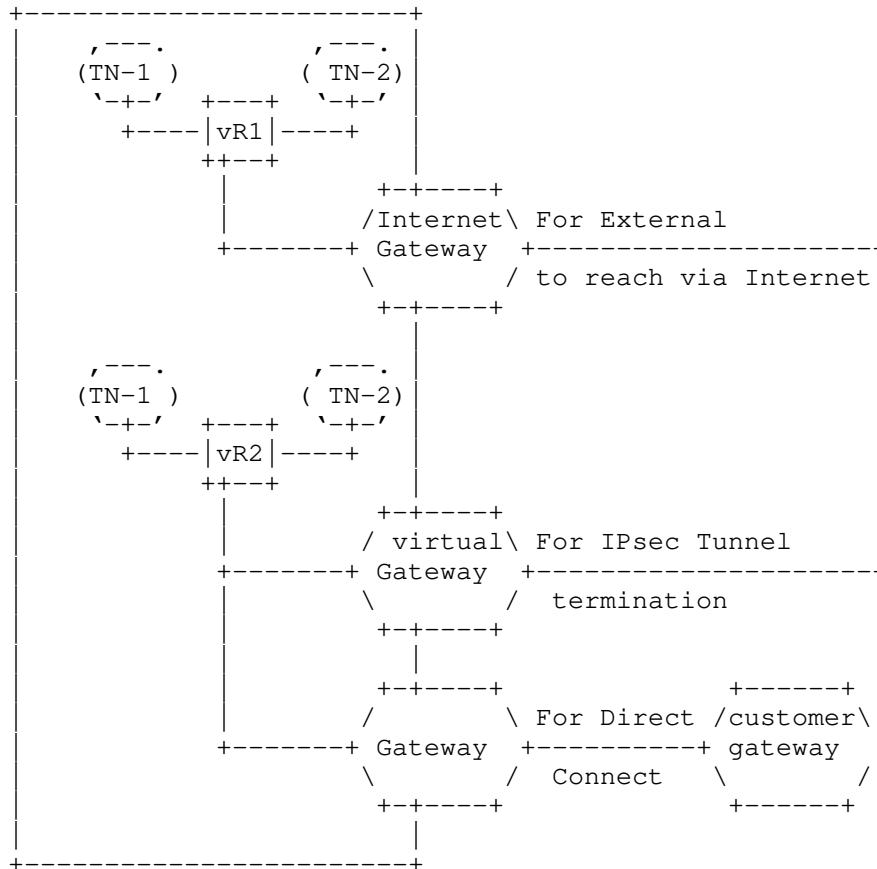


Figure 1: Examples of Multiple Cloud DC connections.

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

RTGWG
Internet-Draft
Intended status: Standards Track
Expires: February 13, 2022

Y. Qu
Futurewei
J. Tantsura
Microsoft
A. Lindem
Cisco
X. Liu
Volta Networks
August 12, 2021

A YANG Data Model for Routing Policy
draft-ietf-rtgwg-policy-model-31

Abstract

This document defines a YANG data model for configuring and managing routing policies in a vendor-neutral way. The model provides a generic routing policy framework which can be extended for specific routing protocols using the YANG 'augment' mechanism.

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 February 13, 2022.

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	2
1.1. Goals and approach	3
2. Terminology and Notation	3
2.1. Tree Diagrams	4
2.2. Prefixes in Data Node Names	4
3. Model overview	5
4. Route policy expression	6
4.1. Defined sets for policy matching	6
4.2. Policy conditions	7
4.3. Policy actions	8
4.4. Policy subroutines	9
5. Policy evaluation	10
6. Applying routing policy	10
7. YANG Module and Tree	11
7.1. Routing Policy Model Tree	11
7.2. Routing policy model	12
8. Security Considerations	32
9. IANA Considerations	34
10. Acknowledgements	34
11. References	34
11.1. Normative references	34
11.2. Informative references	36
Appendix A. Routing protocol-specific policies	36
Appendix B. Policy examples	39
Authors' Addresses	41

1. Introduction

This document describes a YANG [RFC7950] data model for routing policy configuration based on operational usage and best practices in a variety of service provider networks. The model is intended to be vendor-neutral, to allow operators to manage policy configuration consistently in environments with routers supplied by multiple vendors.

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

1.1. Goals and approach

This model does not aim to be feature complete -- it is a subset of the policy configuration parameters available in a variety of vendor implementations, but supports widely used constructs for managing how routes are imported, exported, and modified across different routing protocols. The model development approach has been to examine actual policy configurations in use across several operator networks. Hence, the focus is on enabling policy configuration capabilities and structure that are in wide use.

Despite the differences in details of policy expressions and conventions in various vendor implementations, the model reflects the observation that a relatively simple condition-action approach can be readily mapped to several existing vendor implementations, and also gives operators a familiar and straightforward way to express policy. A side effect of this design decision is that other methods for expressing policies are not considered.

Consistent with the goal to produce a data model that is vendor neutral, only policy expressions that are deemed to be widely available in existing major implementations are included in the model. Those configuration items that are only available from a single implementation are omitted from the model with the expectation they will be available in separate vendor-provided modules that augment the current model.

2. Terminology and Notation

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.

Routing policy: A routing policy defines how routes are imported, exported, modified, and advertised between routing protocol instances or within a single routing protocol instance.

Policy chain: A policy chain is a sequence of policy definitions. They can be referenced from different contexts.

Policy statement: Policy statements consist of a set of conditions and actions (either of which may be empty).

The following terms are defined in [RFC8342]:

- o client

- o server
- o configuration
- o system state
- o operational state
- o intended configuration

The following terms are defined in [RFC7950]:

- o action
- o augment
- o container
- o container with presence
- o data model
- o data node
- o feature
- o leaf
- o list
- o mandatory node
- o module
- o schema tree
- o RPC (Remote Procedure Call) operation

2.1. Tree Diagrams

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

2.2. Prefixes in Data Node Names

In this document, names of data nodes, actions, and other data model objects are often used without a prefix, as long as it is clear from the context in which YANG module each name is defined. Otherwise,

names are prefixed using the standard prefix associated with the corresponding YANG module, as shown in Table 1.

Prefix	YANG module	Reference
if	ietf-interfaces	[RFC8343]
rt	ietf-routing	[RFC8349]
yang	ietf-yang-types	[RFC6991]
inet	ietf-inet-types	[RFC6991]

Table 1: Prefixes and Corresponding YANG Modules

3. Model overview

The routing policy module has three main parts:

- o A generic framework is provided to express policies as sets of related conditions and actions. This includes match sets and actions that are useful across many routing protocols.
- o A structure that allows routing protocol models to add protocol-specific policy conditions and actions through YANG augmentations is also provided. There is a complete example of this for BGP [RFC4271] policies in the proposed vendor-neutral BGP data model [I-D.ietf-idr-bgp-model]. Appendix A provides an example of how an augmentation for BGP policies might be accomplished. Note that this section is not normative as the BGP model is still evolving.
- o Finally, a reusable grouping is defined for attaching import and export rules in the context of routing configuration for different protocols, VRFs, etc. This also enables creation of policy chains and expressing default policy behavior. In this document, policy chains are sequences of policy definitions that are applied in order (described in Section 4).

The module makes use of the standard Internet types, such as IP addresses, autonomous system numbers, etc., defined in RFC 6991 [RFC6991].

4. Route policy expression

Policies are expressed as a sequence of top-level policy definitions each of which consists of a sequence of policy statements. Policy statements in turn consist of simple condition-action tuples. Conditions may include multiple match or comparison operations, and similarly, actions may include multiple changes to route attributes, or indicate a final disposition of accepting or rejecting the route. This structure is shown below.

```

+--rw routing-policy
  +--ro match-modified-attributes?  boolean
  +--rw policy-definitions
    +--rw policy-definition* [name]
      +--rw name                string
      +--rw statements
        +--rw statement* [name]
          +--rw name              string
          +--rw conditions
          |   ...
          +--rw actions
          ...

```

4.1. Defined sets for policy matching

The model provides a collection of generic sets that can be used for matching in policy conditions. These sets are applicable for route selection across multiple routing protocols. They may be further augmented by protocol-specific models which have their own defined sets. The defined sets include:

- o prefix sets - Each prefix set defines a set of IP prefixes, each with an associated IP prefix and netmask range (or exact length).
- o neighbor sets - Each neighbor set defines a set of neighboring nodes by their IP addresses. A neighbor set is used for selecting routes based on the neighbors advertising the routes.
- o tag set - Each tag set defines a set of generic tag values that can be used in matches for filtering routes.

The model structure for defined sets is shown below.

```

+--rw routing-policy
  +--rw defined-sets
    +--rw prefix-sets
      +--rw prefix-set* [name]
        +--rw name          string
        +--rw mode?         enumeration
        +--rw prefixes
          +--rw prefix-list* [ip-prefix mask-length-lower
                               mask-length-upper]
            +--rw ip-prefix      inet:ip-prefix
            +--rw mask-length-lower uint8
            +--rw mask-length-upper uint8
        +--rw neighbor-sets
          +--rw neighbor-set* [name]
            +--rw name          string
            +--rw address*      inet:ip-address
        +--rw tag-sets
          +--rw tag-set* [name]
            +--rw name          string
            +--rw tag-value*    tag-type

```

4.2. Policy conditions

Policy statements consist of a set of conditions and actions (either of which may be empty). Conditions are used to match route attributes against a defined set (e.g., a prefix set), or to compare attributes against a specific value. The default action is to reject-route.

Match conditions may be further modified using the match-set-options configuration which allows network operators to change the behavior of a match. Three options are supported:

- o ALL - match is true only if the given value matches all members of the set.
- o ANY - match is true if the given value matches any member of the set.
- o INVERT - match is true if the given value does not match any member of the given set.

Not all options are appropriate for matching against all defined sets (e.g., match ALL in a prefix set does not make sense). In the model, a restricted set of match options is used where applicable.

Comparison conditions may similarly use options to change how route attributes should be tested, e.g., for equality or inequality, against a given value.

While most policy conditions will be added by individual routing protocol models via augmentation, this routing policy model includes several generic match conditions and the ability to test which protocol or mechanism installed a route (e.g., BGP, IGP, static, etc.). The conditions included in the model are shown below.

```

+--rw routing-policy
  +--rw policy-definitions
    +--rw policy-definition* [name]
      +--rw name                string
      +--rw statements
        +--rw statement* [name]
          +--rw conditions
            +--rw call-policy?
            +--rw source-protocol?
            +--rw match-interface
            |   +--rw interface?
            +--rw match-prefix-set
            |   +--rw prefix-set?
            |   +--rw match-set-options?
            +--rw match-neighbor-set
            |   +--rw neighbor-set?
            +--rw match-tag-set
            |   +--rw tag-set?
            |   +--rw match-set-options?
            +--rw match-route-type* identityref
            +--rw route-type*

```

4.3. Policy actions

When policy conditions are satisfied, policy actions are used to set various attributes of the route being processed, or to indicate the final disposition of the route, i.e., accept or reject.

Similar to policy conditions, the routing policy model includes generic actions in addition to the basic route disposition actions. These are shown below.

```

+--rw routing-policy
  +--rw policy-definitions
    +--rw policy-definition* [name]
      +--rw statements
        +--rw statement* [name]
          +--rw actions
            +--rw policy-result?    policy-result-type
            +--rw set-metric
              | +--rw metric-modification?
              | | metric-modification-type
              | +--rw metric?          uint32
            +--rw set-metric-type
              | +--rw metric-type?    identityref
            +--rw set-route-level
              | +--rw route-level?   identityref
            +--rw set-route-preference? uint16
            +--rw set-tag?            tag-type
            +--rw set-application-tag? tag-type

```

4.4. Policy subroutines

Policy 'subroutines' (or nested policies) are supported by allowing policy statement conditions to reference other policy definitions using the call-policy configuration. Called policies apply their conditions and actions before returning to the calling policy statement and resuming evaluation. The outcome of the called policy affects the evaluation of the calling policy. If the called policy results in an accept-route, then the subroutine returns an effective Boolean true value to the calling policy. For the calling policy, this is equivalent to a condition statement evaluating to a true value and evaluation of the policy continues (see Section 5). Note that the called policy may also modify attributes of the route in its action statements. Similarly, a reject-route action returns false and the calling policy evaluation will be affected accordingly. When the end of the subroutine policy statements is reached, the default route disposition action is returned (i.e., Boolean false for reject-route). Consequently, a subroutine cannot explicitly accept or reject a route. Rather, the called policy returns Boolean true if its outcome is accept-route or Boolean false if its outcome is reject-route. Route acceptance or rejection is solely determined by the top-level policy.

Note that the called policy may itself call other policies (subject to implementation limitations). The model does not prescribe a nesting depth because this varies among implementations. For example, an implementation may only support a single level of subroutine recursion. As with any routing policy construction, care must be taken with nested policies to ensure that the effective

return value results in the intended behavior. Nested policies are a convenience in many routing policy constructions but creating policies nested beyond a small number of levels (e.g., 2-3) is discouraged. Also, implementations MUST validate to ensure that there is no recursion among nested routing policies.

5. Policy evaluation

Evaluation of each policy definition proceeds by evaluating its individual policy statements in order that they are defined. When all the condition statements in a policy statement are satisfied, the corresponding action statements are executed. If the actions include either accept-route or reject-route actions, evaluation of the current policy definition stops, and no further policy statement is evaluated. If there are multiple policies in the policy chain, subsequent policies are not evaluated. Policy chains are sequences of policy definitions (as described in Section 4).

If the conditions are not satisfied, then evaluation proceeds to the next policy statement. If none of the policy statement conditions are satisfied, then evaluation of the current policy definition stops, and the next policy definition in the chain is evaluated. When the end of the policy chain is reached, the default route disposition action is performed (i.e., reject-route unless an alternate default action is specified for the chain).

Whether the route's pre-policy attributes are used for testing policy statement conditions is dependent on the implementation specific value of the match-modified-attributes leaf. If match-modified-attributes is false and actions modify route attributes, these modifications are not used for policy statement conditions. Conversely, if match-modified-attributes is true and actions modify the policy application-specific attributes, the attributes as modified by the policy are used for policy condition statements.

6. Applying routing policy

Routing policy is applied by defining and attaching policy chains in various routing contexts. Policy chains are sequences of policy definitions (described in Section 4). They can be referenced from different contexts. For example, a policy chain could be associated with a routing protocol and used to control its interaction with its protocol peers. Or it could be used to control the interaction between a routing protocol and the local routing information base. A policy chain has an associated direction (import or export), with respect to the context in which it is referenced.

The routing policy model defines an apply-policy grouping that can be imported and used by other models. As shown below, it allows definition of import and export policy chains, as well as specifying the default route disposition to be used when no policy definition in the chain results in a final decision.

```
+--rw apply-policy
|   +--rw import-policy*
|   +--rw default-import-policy?   default-policy-type
|   +--rw export-policy*
|   +--rw default-export-policy?   default-policy-type
```

The default policy defined by the model is to reject the route for both import and export policies.

7. YANG Module and Tree

7.1. Routing Policy Model Tree

The tree of the routing policy model is shown below.

```
module: ietf-routing-policy
rw routing-policy
+--rw defined-sets
|   +--rw prefix-sets
|   |   +--rw prefix-set* [name mode]
|   |   |   +--rw name          string
|   |   |   +--rw mode          enumeration
|   |   |   +--rw prefixes
|   |   |   |   +--rw prefix-list* [ip-prefix mask-length-lower
|   |   |   |   |   mask-length-upper]
|   |   |   |   |   +--rw ip-prefix          inet:ip-prefix
|   |   |   |   |   +--rw mask-length-lower   uint8
|   |   |   |   |   +--rw mask-length-upper   uint8
|   |   +--rw neighbor-sets
|   |   |   +--rw neighbor-set* [name]
|   |   |   |   +--rw name          string
|   |   |   |   +--rw address*      inet:ip-address
|   |   +--rw tag-sets
|   |   |   +--rw tag-set* [name]
|   |   |   |   +--rw name          string
|   |   |   |   +--rw tag-value*    tag-type
|   +--rw policy-definitions
|   |   +--ro match-modified-attributes?   boolean
|   |   +--rw policy-definition* [name]
|   |   |   +--rw name          string
|   |   |   +--rw statements
|   |   |   |   +--rw statement* [name]
```



```

+--rw name                string
+--rw conditions
|   +--rw call-policy?      -> ../../../../..
|                           /policy-definitions
|                           /policy-definition/name
|   +--rw source-protocol?  identityref
|   +--rw match-interface
|   |   +--rw interface?    -> /if:interfaces/interface
|   |                       /name
|   +--rw match-prefix-set
|   |   +--rw prefix-set?   -> ../../../../..
|   |                       /defined-sets/prefix-sets
|   |                       /prefix-set/name
|   |   +--rw match-set-options? match-set-options-type
|   +--rw match-neighbor-set
|   |   +--rw neighbor-set? -> ../../../../..
|   |                       /defined-sets/neighbor-sets
|   |                       /neighbor-set/name
|   +--rw match-tag-set
|   |   +--rw tag-set?      -> ../../../../..
|   |                       /defined-sets/tag-sets
|   |                       /tag-set/name
|   |   +--rw match-set-options? match-set-options-type
|   +--rw match-route-type* identityref
+--rw actions
|   +--rw policy-result?    policy-result-type
|   +--rw set-metric
|   |   +--rw metric-modification? metric-modification-type
|   |   +--rw metric?       uint32
|   +--rw set-metric-type
|   |   +--rw metric-type?   identityref
|   +--rw set-route-level
|   |   +--rw route-level?   identityref
|   +--rw set-route-preference? uint16
|   +--rw set-tag?          tag-type
|   +--rw set-application-tag? tag-type

```

7.2. Routing policy model

The following RFCs are not referenced in the document text but are referenced in the `ietf-routing-policy.yang` module: [RFC2328], [RFC3101], [RFC5130], [RFC5302], [RFC6991], and [RFC8343].

```

<CODE BEGINS> file "ietf-routing-policy@2021-08-12.yang"
module ietf-routing-policy {

  yang-version "1.1";

```

```
namespace "urn:ietf:params:xml:ns:yang:ietf-routing-policy";
prefix rt-pol;

import ietf-inet-types {
  prefix "inet";
  reference
    "RFC 6991: Common YANG Data Types";
}

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 (NMDA Version)";
}

import ietf-routing {
  prefix "rt";
  reference
    "RFC 8349: A YANG Data Model for Routing
      Management (NMDA Version)";
}

organization
  "IETF RTGWG - Routing Area Working Group";
contact
  "WG Web:    <https://datatracker.ietf.org/wg/rtgw/>
  WG List:    <mailto: rtgw@ietf.org>

  Editor:     Yingzhen Qu
               <mailto: yingzhen.qu@futurewei.com>
               Jeff Tantsura
               <mailto: jefftant.ietf@gmail.com>
               Acee Lindem
               <mailto: acee@cisco.com>
               Xufeng Liu
               <mailto: xufeng.liu.ietf@gmail.com>";

description
  "This module describes a YANG model for routing policy
  configuration. It is a limited subset of all of the policy
  configuration parameters available in the variety of vendor
```

implementations, but supports widely used constructs for managing how routes are imported, exported, modified and advertised across different routing protocol instances or within a single routing protocol instance. This module is intended to be used in conjunction with routing protocol configuration modules (e.g., BGP) defined in other models.

This YANG module conforms to the Network Management Datastore Architecture (NMDA), as described in RFC 8342.

Copyright (c) 2021 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 (<https://trustee.ietf.org/license-info>).

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

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 (RFC 2119) (RFC 8174) when, and only when, they appear in all capitals, as shown here.";

reference "RFC XXXX: A YANG Data Model for Routing Policy.";

```
revision "2021-08-12" {
  description
    "Initial revision.";
  reference
    "RFC XXXX: A YANG Data Model for Routing Policy Management.";
}
```

```
/* Identities */
```

```
identity metric-type {
  description
    "Base identity for route metric types.";
}
```

```
identity ospf-type-1-metric {
  base metric-type;
  description
```

```
        "Identity for the OSPF type 1 external metric types. It
        is only applicable to OSPF routes.";
    reference
        "RFC 2328: OSPF Version 2";
}

identity ospf-type-2-metric {
    base metric-type;
    description
        "Identity for the OSPF type 2 external metric types. It
        is only applicable to OSPF routes.";
    reference
        "RFC 2328: OSPF Version 2";
}

identity isis-internal-metric {
    base metric-type;
    description
        "Identity for the IS-IS internal metric types. It is only
        applicable to IS-IS routes.";
    reference
        "RFC 5302: Domain-Wide Prefix Distribution with
        Two-Level IS-IS";
}

identity isis-external-metric {
    base metric-type;
    description
        "Identity for the IS-IS external metric types. It is only
        applicable to IS-IS routes.";
    reference
        "RFC 5302: Domain-Wide Prefix Distribution with
        Two-Level IS-IS";
}

identity route-level {
    description
        "Base identity for route import level.";
}

identity ospf-normal {
    base route-level;
    description
        "Identity for OSPF importation into normal areas
        It is only applicable to routes imported
        into the OSPF protocol.";
    reference
        "RFC 2328: OSPF Version 2";
}
```

```
}

identity ospf-nssa-only {
  base route-level;
  description
    "Identity for the OSPF Not-So-Stubby Area (NSSA) area
    importation. It is only applicable to routes imported
    into the OSPF protocol.";
  reference
    "RFC 3101: The OSPF Not-So-Stubby Area (NSSA) Option";
}

identity ospf-normal-nssa {
  base route-level;
  description
    "Identity for OSPF importation into both normal and NSSA
    areas, it is only applicable to routes imported into
    the OSPF protocol.";
  reference
    "RFC 3101: The OSPF Not-So-Stubby Area (NSSA) Option";
}

identity isis-level-1 {
  base route-level;
  description
    "Identity for IS-IS Level 1 area importation. It is only
    applicable to routes imported into the IS-IS protocol.";
  reference
    "RFC 5302: Domain-Wide Prefix Distribution with
    Two-Level IS-IS";
}

identity isis-level-2 {
  base route-level;
  description
    "Identity for IS-IS Level 2 area importation. It is only
    applicable to routes imported into the IS-IS protocol.";
  reference
    "RFC 5302: Domain-Wide Prefix Distribution with
    Two-Level IS-IS";
}

identity isis-level-1-2 {
  base route-level;
  description
    "Identity for IS-IS importation into both Level 1 and Level 2
    areas. It is only applicable to routes imported into the IS-IS
    protocol.";
```

```
reference
  "RFC 5302: Domain-Wide Prefix Distribution with
    Two-Level IS-IS";
}

identity proto-route-type {
  description
    "Base identity for route type within a protocol.";
}

identity isis-level-1-type {
  base proto-route-type;
  description
    "Identity for IS-IS Level 1 route type. It is only
      applicable to IS-IS routes.";
  reference
    "RFC 5302: Domain-Wide Prefix Distribution with
      Two-Level IS-IS";
}

identity isis-level-2-type {
  base proto-route-type;
  description
    "Identity for IS-IS Level 2 route type. It is only
      applicable to IS-IS routes.";
  reference
    "RFC 5302: Domain-Wide Prefix Distribution with
      Two-Level IS-IS";
}

identity ospf-internal-type {
  base proto-route-type;
  description
    "Identity for OSPF intra-area or inter-area route type.
      It is only applicable to OSPF routes.";
  reference
    "RFC 2328: OSPF Version 2";
}

identity ospf-external-type {
  base proto-route-type;
  description
    "Identity for OSPF external type 1/2 route type.
      It is only applicable to OSPF routes.";
  reference
    "RFC 2328: OSPF Version 2";
}
```

```
identity ospf-external-t1-type {
  base ospf-external-type;
  description
    "Identity for OSPF external type 1 route type.
    It is only applicable to OSPF routes.";
  reference
    "RFC 2328: OSPF Version 2";
}

identity ospf-external-t2-type {
  base ospf-external-type;
  description
    "Identity for OSPF external type 2 route type.
    It is only applicable to OSPF routes.";
  reference
    "RFC 2328: OSPF Version 2";
}

identity ospf-nssa-type {
  base proto-route-type;
  description
    "Identity for OSPF NSSA type 1/2 route type.
    It is only applicable to OSPF routes.";
  reference
    "RFC 3101: The OSPF Not-So-Stubby Area (NSSA) Option";
}

identity ospf-nssa-t1-type {
  base ospf-nssa-type;
  description
    "Identity for OSPF NSSA type 1 route type.
    It is only applicable to OSPF routes.";
  reference
    "RFC 3101: The OSPF Not-So-Stubby Area (NSSA) Option";
}

identity ospf-nssa-t2-type {
  base ospf-nssa-type;
  description
    "Identity for OSPF NSSA type 2 route type.
    It is only applicable to OSPF routes.";
  reference
    "RFC 3101: The OSPF Not-So-Stubby Area (NSSA) Option";
}

identity bgp-internal {
  base proto-route-type;
  description
```

```
    "Identity for routes learned from internal BGP (IBGP).  
    It is only applicable to BGP routes.";  
reference  
    "RFC 4271: A Border Gateway Protocol 4 (BGP-4)";  
}  
  
identity bgp-external {  
    base proto-route-type;  
    description  
        "Identity for routes learned from external BGP (EBGP).  
        It is only applicable to BGP routes.";  
    reference  
        "RFC 4271: A Border Gateway Protocol 4 (BGP-4)";  
}  
  
/* Type Definitions */  
  
typedef default-policy-type {  
    type enumeration {  
        enum accept-route {  
            description  
                "Default policy to accept the route.";  
        }  
        enum reject-route {  
            description  
                "Default policy to reject the route.";  
        }  
    }  
    description  
        "Type used to specify route disposition in  
        a policy chain. This typedef is used in  
        the default import and export policy.";  
}  
  
typedef policy-result-type {  
    type enumeration {  
        enum accept-route {  
            description  
                "Policy accepts the route.";  
        }  
        enum reject-route {  
            description  
                "Policy rejects the route.";  
        }  
    }  
    description  
        "Type used to specify route disposition in  
        a policy chain.";
```



```
}

typedef tag-type {
  type union {
    type uint32;
    type yang:hex-string;
  }
  description
    "Type for expressing route tags on a local system,
    including IS-IS and OSPF; may be expressed as either decimal
    or hexadecimal integer.";
  reference
    "RFC 2328: OSPF Version 2
    RFC 5130: A Policy Control Mechanism in IS-IS Using
    Administrative Tags";
}

typedef match-set-options-type {
  type enumeration {
    enum any {
      description
        "Match is true if given value matches any member
        of the defined set.";
    }
    enum all {
      description
        "Match is true if given value matches all
        members of the defined set.";
    }
    enum invert {
      description
        "Match is true if given value does not match any
        member of the defined set.";
    }
  }
  default any;
  description
    "Options that govern the behavior of a match statement. The
    default behavior is any, i.e., the given value matches any
    of the members of the defined set.";
}

typedef metric-modification-type {
  type enumeration {
    enum set-metric {
      description
        "Set the metric to the specified value.";
    }
  }
}
```

```
enum add-metric {
    description
        "Add the specified value to the existing metric.
        If the result overflows the maximum metric
        (0xffffffff), set the metric to the maximum.";
}
enum subtract-metric {
    description
        "Subtract the specified value from the existing metric. If
        the result is less than 0, set the metric to 0.";
}
}
description
    "Type used to specify how to set the metric given the
    specified value.";
}

/* Groupings */

grouping prefix {
    description
        "Configuration data for a prefix definition.

        The combination of mask-length-lower and mask-length-upper
        define a range for the mask length, or single 'exact'
        length if mask-length-lower and mask-length-upper are
        equal.

        Example: 192.0.2.0/24 through 192.0.2.0/26 would be
        expressed as prefix: 192.0.2.0/24,
                mask-length-lower=24,
                mask-length-upper=26

        Example: 192.0.2.0/24 (an exact match) would be
        expressed as prefix: 192.0.2.0/24,
                mask-length-lower=24,
                mask-length-upper=24

        Example: 2001:DB8::/32 through 2001:DB8::/64 would be
        expressed as prefix: 2001:DB8::/32,
                mask-length-lower=32,
                mask-length-upper=64";

    leaf ip-prefix {
        type inet:ip-prefix;
        mandatory true;
        description
            "The IP prefix represented as an IPv6 or IPv4 network
```

```
        number followed by a prefix length with an intervening
        slash character as a delimiter. All members of the
        prefix-set MUST be of the same address family as the
        prefix-set mode.";
    }

    leaf mask-length-lower {
        type uint8 {
            range "0..128";
        }
        description
            "Mask length range lower bound. It MUST NOT be less than
            the prefix length defined in ip-prefix.";
    }
    leaf mask-length-upper {
        type uint8 {
            range "1..128";
        }
        must "../mask-length-upper >= ../mask-length-lower" {
            error-message "The upper bound MUST NOT be less"
                + "than lower bound.";
        }
        description
            "Mask length range upper bound. It MUST NOT be less than
            lower bound.";
    }
}

grouping match-set-options-group {
    description
        "Grouping containing options relating to how a particular set
        will be matched.";

    leaf match-set-options {
        type match-set-options-type;
        description
            "Optional parameter that governs the behavior of the
            match operation.";
    }
}

grouping match-set-options-restricted-group {
    description
        "Grouping for a restricted set of match operation
        modifiers.";

    leaf match-set-options {
        type match-set-options-type {
```

```
    enum any {
      description
        "Match is true if given value matches any
        member of the defined set.";
    }
    enum invert {
      description
        "Match is true if given value does not match
        any member of the defined set.";
    }
  }
  description
    "Optional parameter that governs the behavior of the
    match operation. This leaf only supports matching on
    'any' member of the set or 'invert' the match.
    Matching on 'all' is not supported.";
}

grouping apply-policy-group {
  description
    "Top level container for routing policy applications. This
    grouping is intended to be used in routing models where
    needed.";

  container apply-policy {
    description
      "Anchor point for routing policies in the model.
      Import and export policies are with respect to the local
      routing table, i.e., export (send) and import (receive),
      depending on the context.";

    leaf-list import-policy {
      type leafref {
        path "/rt-pol:routing-policy/rt-pol:policy-definitions/" +
          "rt-pol:policy-definition/rt-pol:name";
        require-instance true;
      }
      ordered-by user;
      description
        "List of policy names in sequence to be applied on
        receiving redistributed routes from another routing protocol
        or receiving a routing update in the current context, e.g.,
        for the current peer group, neighbor, address family, etc.";
    }

    leaf default-import-policy {
      type default-policy-type;
    }
  }
}
```

```
    default reject-route;
    description
        "Explicitly set a default policy if no policy definition
        in the import policy chain is satisfied.";
}

leaf-list export-policy {
    type leafref {
        path "/rt-pol:routing-policy/rt-pol:policy-definitions/" +
            "rt-pol:policy-definition/rt-pol:name";
        require-instance true;
    }
    ordered-by user;
    description
        "List of policy names in sequence to be applied on
        redistributing routes from one routing protocol to another
        or sending a routing update in the current context, e.g.,
        for the current peer group, neighbor, address family, etc.";
}

leaf default-export-policy {
    type default-policy-type;
    default reject-route;
    description
        "Explicitly set a default policy if no policy definition
        in the export policy chain is satisfied.";
}
}

container routing-policy {
    description
        "Top-level container for all routing policy.";

    container defined-sets {
        description
            "Predefined sets of attributes used in policy match
            statements.";

        container prefix-sets {
            description
                "Data definitions for a list of IPv4 or IPv6
                prefixes which are matched as part of a policy.";
            list prefix-set {
                key "name mode";
                description
                    "List of the defined prefix sets";
            }
        }
    }
}
```

```
leaf name {
  type string;
  description
    "Name of the prefix set -- this is used as a label to
    reference the set in match conditions.";
}

leaf mode {
  type enumeration {
    enum ipv4 {
      description
        "Prefix set contains IPv4 prefixes only.";
    }
    enum ipv6 {
      description
        "Prefix set contains IPv6 prefixes only.";
    }
  }
  description
    "Indicates the mode of the prefix set, in terms of
    which address families (IPv4 or IPv6) are present.
    The mode provides a hint, all prefixes MUST be of
    the indicated type. The device MUST validate that
    all prefixes and reject the configuration if there
    is a discrepancy.";
}

container prefixes {
  description
    "Container for the list of prefixes in a policy
    prefix list. Since individual prefixes do not have
    unique actions, the order in which the prefix in
    prefix-list are matched has no impact on the outcome
    and is left to the implementation. A given prefix-set
    condition is satisfied if the input prefix matches
    any of the prefixes in the prefix-set.";

  list prefix-list {
    key "ip-prefix mask-length-lower mask-length-upper";
    description
      "List of prefixes in the prefix set.";

    uses prefix;
  }
}
}
```

```
container neighbor-sets {
  description
    "Data definition for a list of IPv4 or IPv6
    neighbors which can be matched in a routing policy.";

  list neighbor-set {
    key "name";
    description
      "List of defined neighbor sets for use in policies.";

    leaf name {
      type string;
      description
        "Name of the neighbor set -- this is used as a label
        to reference the set in match conditions.";
    }

    leaf-list address {
      type inet:ip-address;
      description
        "List of IP addresses in the neighbor set.";
    }
  }
}

container tag-sets {
  description
    "Data definitions for a list of tags which can
    be matched in policies.";

  list tag-set {
    key "name";
    description
      "List of tag set definitions.";

    leaf name {
      type string;
      description
        "Name of the tag set -- this is used as a label to
        reference the set in match conditions.";
    }

    leaf-list tag-value {
      type tag-type;
      description
        "Value of the tag set member.";
    }
  }
}
```

```
    }  
  }  
  
  container policy-definitions {  
    description  
      "Enclosing container for the list of top-level policy  
      definitions.";  
  
    leaf match-modified-attributes {  
      type boolean;  
      config false;  
      description  
        "This boolean value dictates whether matches are performed  
        on the actual route attributes or route attributes  
        modified by policy statements preceding the match.";  
    }  
  
    list policy-definition {  
      key "name";  
      description  
        "List of top-level policy definitions, keyed by unique  
        name. These policy definitions are expected to be  
        referenced (by name) in policy chains specified in  
        import or export configuration statements.";  
  
      leaf name {  
        type string;  
        description  
          "Name of the top-level policy definition -- this name  
          is used in references to the current policy.";  
      }  
  
      container statements {  
        description  
          "Enclosing container for policy statements.";  
  
        list statement {  
          key "name";  
          ordered-by user;  
          description  
            "Policy statements group conditions and actions  
            within a policy definition. They are evaluated in  
            the order specified.";  
  
          leaf name {  
            type string;  
            description  
              "Name of the policy statement.";  
          }  
        }  
      }  
    }  
  }  
}
```



```
}

container conditions {
  description
    "Condition statements for the current policy
    statement.";

  leaf call-policy {
    type leafref {
      path "../..../..../.." +
        "rt-pol:policy-definitions/" +
        "rt-pol:policy-definition/rt-pol:name";
      require-instance true;
    }
    description
      "Applies the statements from the specified policy
      definition and then returns control to the current
      policy statement. Note that the called policy
      may itself call other policies (subject to
      implementation limitations). This is intended to
      provide a policy 'subroutine' capability. The
      called policy SHOULD contain an explicit or a
      default route disposition that returns an
      effective true (accept-route) or false
      (reject-route), otherwise the behavior may be
      ambiguous.";
  }

  leaf source-protocol {
    type identityref {
      base rt:control-plane-protocol;
    }
    description
      "Condition to check the protocol / method used to
      install the route into the local routing table.";
  }

  container match-interface {
    leaf interface {
      type leafref {
        path "/if:interfaces/if:interface/if:name";
      }
      description
        "Reference to a base interface.";
    }
    description
      "Container for interface match conditions";
  }
}
```

```
container match-prefix-set {
  leaf prefix-set {
    type leafref {
      path "../../../../../defined-sets/" +
        "prefix-sets/prefix-set/name";
    }
    description
      "References a defined prefix set.";
  }
  uses match-set-options-restricted-group;

  description
    "Match a referenced prefix-set according to the
    logic defined in the match-set-options leaf.";
}

container match-neighbor-set {
  leaf neighbor-set {
    type leafref {
      path "../../../../../defined-sets/" +
        "neighbor-sets/neighbor-set/name";
      require-instance true;
    }
    description
      "References a defined neighbor set.";
  }

  description
    "Match a referenced neighbor set.";
}

container match-tag-set {
  leaf tag-set {
    type leafref {
      path "../../../../../defined-sets/" +
        "tag-sets/tag-set/name";
      require-instance true;
    }
    description
      "References a defined tag set.";
  }
  uses match-set-options-group;

  description
    "Match a referenced tag set according to the logic
    defined in the match-set-options leaf.";
}
```

```
    container match-route-type {
      description
        "This container provides route-type match condition";

      leaf-list route-type {
        type identityref {
          base proto-route-type;
        }
        description
          "Condition to check the protocol-specific type
           of route. This is normally used during route
           importation to select routes or to set protocol
           specific attributes based on the route type.";
      }
    }
  }

  container actions {
    description
      "Top-level container for policy action
       statements.";
    leaf policy-result {
      type policy-result-type;
      default reject-route;
      description
        "Select the final disposition for the route,
         either accept or reject.";
    }
    container set-metric {
      leaf metric-modification {
        type metric-modification-type;
        description
          "Indicates how to modify the metric.";
      }
      leaf metric {
        type uint32;
        description
          "Metric value to set, add, or subtract.";
      }
      description
        "Set the metric for the route.";
    }
    container set-metric-type {
      leaf metric-type {
        type identityref {
          base metric-type;
        }
        description

```



```
}  
}  
<CODE ENDS>
```

8. 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 (NACM) [RFC8341] provides the means to restrict access for particular NETCONF or RESTCONF users to a pre-configured 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:

```
/routing-policy/defined-sets/prefix-sets -- Modification to  
prefix-sets could result in a Denial-of-Service (DoS) attack. An  
attacker may try to modify prefix-sets and redirect or drop  
traffic. Redirection of traffic could be used as part of a more  
elaborate attack to either collect sensitive information or  
masquerade a service. Additionally, a control-plane DoS attack  
could be accomplished by allowing a large number of routes to be  
leaked into a routing protocol domain (e.g., BGP).
```

```
/routing-policy/defined-sets/neighbor-sets -- Modification to the  
neighbor-sets could be used to mount a DoS attack or more  
elaborate attack as with prefix-sets. For example, a DoS attack  
could be mounted by changing the neighbor-set from which routes  
are accepted.
```

```
/routing-policy/defined-sets/tag-sets -- Modification to the tag-  
sets could be used to mount a DoS attack. Routes with certain  
tags might be redirected or dropped. The implications are similar  
to prefix-sets and neighbor-sets. However, the attack may be more  
difficult to detect as the routing policy usage of route tags and
```

intent must be understood to recognize the breach. Conversely, the implications of prefix-set or neighbor set modification are easier to recognize.

```
/routing-policy/policy-definitions/policy-definition
/statements/statement/conditions -- Modification to the conditions
could be used to mount a DoS attack or other attack. An attacker
may change a policy condition and redirect or drop traffic. As
with prefix-sets, neighbor-sets, or tag-sets, traffic redirection
could be used as part of a more elaborate attack.
```

```
/routing-policy/policy-definitions/policy-definition
/statements/statement/actions -- Modification to actions could be
used to mount a DoS attack or other attack. Traffic may be
redirected or dropped. As with prefix-sets, neighbor-sets, or
tag-sets, traffic redirection could be used as part of a more
elaborate attack. Additionally, route attributes may be changed
to mount a second-level attack that is more difficult to detect.
```

Some of the readable data nodes in the YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control read access (e.g., via get, get-config, or notification) to these data nodes. These are the subtrees and data nodes and their sensitivity/vulnerability:

```
/routing-policy/defined-sets/prefix-sets -- Knowledge of these
data nodes can be used to ascertain which local prefixes are
susceptible to a Denial-of-Service (DoS) attack.
```

```
/routing-policy/defined-sets/prefix-sets -- Knowledge of these
data nodes can be used to ascertain local neighbors against whom
to mount a Denial-of-Service (DoS) attack.
```

```
/routing-policy/policy-definitions/policy-definition /statements/
-- Knowledge of these data nodes can be used to attack the local
router with a Denial-of-Service (DoS) attack. Additionally,
policies and their attendant conditions and actions should be
considered proprietary and disclosure could be used to ascertain
partners, customers, and supplies. Furthermore, the policies
themselves could represent intellectual property and disclosure
could diminish their corresponding business advantage.
```

Routing policy configuration has a significant impact on network operations, and, as such, other YANG models that reference routing policies are also susceptible to vulnerabilities relating the YANG data nodes specified above.

9. 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-routing-policy
Registrant Contact: The IESG.
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-routing-policy
namespace: urn:ietf:params:xml:ns:yang:ietf-routing-policy
prefix: rt-pol
reference: RFC XXXX
```

10. Acknowledgements

The routing policy module defined in this document is based on the OpenConfig route policy model. The authors would like to thank to OpenConfig for their contributions, especially Anees Shaikh, Rob Shakir, Kevin D'Souza, and Chris Chase.

The authors are grateful for valuable contributions to this document and the associated models from: Ebben Aires, Luyuan Fang, Josh George, Stephane Litkowski, Ina Minei, Carl Moberg, Eric Osborne, Steve Padgett, Juergen Schoenwaelder, Jim Uttaro, Russ White, and John Heasley.

Thanks to Mahesh Jethanandani, John Scudder, Chris Bowers and Tom Petch for their reviews and comments.

11. References

11.1. 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>>.
- [RFC2328] Moy, J., "OSPF Version 2", STD 54, RFC 2328, DOI 10.17487/RFC2328, April 1998, <<https://www.rfc-editor.org/info/rfc2328>>.

- [RFC3101] Murphy, P., "The OSPF Not-So-Stubby Area (NSSA) Option", RFC 3101, DOI 10.17487/RFC3101, January 2003, <<https://www.rfc-editor.org/info/rfc3101>>.
- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <<https://www.rfc-editor.org/info/rfc3688>>.
- [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>>.
- [RFC5130] Previdi, S., Shand, M., Ed., and C. Martin, "A Policy Control Mechanism in IS-IS Using Administrative Tags", RFC 5130, DOI 10.17487/RFC5130, February 2008, <<https://www.rfc-editor.org/info/rfc5130>>.
- [RFC5302] Li, T., Smit, H., and T. Przygienda, "Domain-Wide Prefix Distribution with Two-Level IS-IS", RFC 5302, DOI 10.17487/RFC5302, October 2008, <<https://www.rfc-editor.org/info/rfc5302>>.
- [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>>.
- [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>>.
- [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>>.
- [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>>.

- [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>>.
- [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>>.
- [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>>.
- [RFC8349] Lhotka, L., Lindem, A., and Y. Qu, "A YANG Data Model for Routing Management (NMDA Version)", RFC 8349, DOI 10.17487/RFC8349, March 2018, <<https://www.rfc-editor.org/info/rfc8349>>.
- [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>>.

11.2. Informative references

- [I-D.ietf-idr-bgp-model]
Jethanandani, M., Patel, K., Hares, S., and J. Haas, "BGP YANG Model for Service Provider Networks", draft-ietf-idr-bgp-model-11 (work in progress), July 2021.

Appendix A. Routing protocol-specific policies

Routing models that require the ability to apply routing policy may augment the routing policy model with protocol or other specific policy configuration. The routing policy model assumes that additional defined sets, conditions, and actions may all be added by other models.

The example below provides an illustration of how another data model can augment parts of this routing policy data model. It uses

specific examples from draft-ietf-idr-bgp-model-09 to show in a concrete manner how the different pieces fit together. This example is not normative with respect to [I-D.ietf-idr-bgp-model]. The model similarly augments BGP-specific conditions and actions in the corresponding sections of the routing policy model. In the example below, the XPath prefix "bp:" specifies import from the ietf-bgp-policy sub-module and the XPath prefix "bt:" specifies import from the ietf-bgp-types sub-module [I-D.ietf-idr-bgp-model].

```

module: ietf-routing-policy
+--rw routing-policy
|   +--rw defined-sets
|   |   +--rw prefix-sets
|   |   |   +--rw prefix-set* [name]
|   |   |   |   +--rw name          string
|   |   |   |   +--rw mode?         enumeration
|   |   |   |   +--rw prefixes
|   |   |   |   |   +--rw prefix-list* [ip-prefix mask-length-lower
|   |   |   |   |   |   mask-length-upper]
|   |   |   |   |   +--rw ip-prefix          inet:ip-prefix
|   |   |   |   |   +--rw mask-length-lower  uint8
|   |   |   |   |   +--rw mask-length-upper  uint8
|   |   +--rw neighbor-sets
|   |   |   +--rw neighbor-set* [name]
|   |   |   |   +--rw name          string
|   |   |   |   +--rw address*      inet:ip-address
|   |   +--rw tag-sets
|   |   |   +--rw tag-set* [name]
|   |   |   |   +--rw name          string
|   |   |   |   +--rw tag-value*    tag-type
|   |   +--rw bp:bp-defined-sets
|   |   |   +--rw bp:community-sets
|   |   |   |   +--rw bp:community-set* [name]
|   |   |   |   |   +--rw bp:name          string
|   |   |   |   |   +--rw bp:member*      union
|   |   |   +--rw bp:ext-community-sets
|   |   |   |   +--rw bp:ext-community-set* [name]
|   |   |   |   |   +--rw bp:name          string
|   |   |   |   |   +--rw bp:member*      union
|   |   +--rw bp:as-path-sets
|   |   |   +--rw bp:as-path-set* [name]
|   |   |   |   +--rw bp:name          string
|   |   |   |   +--rw bp:member*      string
|   +--rw policy-definitions
|   |   +--ro match-modified-attributes?  boolean
|   |   +--rw policy-definition* [name]
|   |   |   +--rw name          string
|   |   |   +--rw statements

```

```

+--rw statement* [name]
  +--rw name          string
  +--rw conditions
    +--rw call-policy?
    +--rw source-protocol?          identityref
    +--rw match-interface
    |   +--rw interface?
    +--rw match-prefix-set
    |   +--rw prefix-set?          prefix-set/name
    |   +--rw match-set-options?  match-set-options-type
    +--rw match-neighbor-set
    |   +--rw neighbor-set?
    +--rw match-tag-set
    |   +--rw tag-set?
    |   +--rw match-set-options?  match-set-options-type
    +--rw match-route-type*  identityref
    +--rw bp:bgp-conditions
      +--rw bp:med-eq?          uint32
      +--rw bp:origin-eq?      bt:bgp-origin-attr-type
      +--rw bp:next-hop-in*    inet:ip-address-no-zone
      +--rw bp:afi-safi-in*    identityref
      +--rw bp:local-pref-eq?  uint32
      +--rw bp:route-type?    enumeration
      +--rw bp:community-count
      +--rw bp:as-path-length
      +--rw bp:match-community-set
      |   +--rw bp:community-set?
      |   +--rw bp:match-set-options?
      +--rw bp:match-ext-community-set
      |   +--rw bp:ext-community-set?
      |   +--rw bp:match-set-options?
      +--rw bp:match-as-path-set
      |   +--rw bp:as-path-set?
      |   +--rw bp:match-set-options?
    +--rw actions
      +--rw policy-result?      policy-result-type
      +--rw set-metric
      |   +--rw metric-modification?
      |   +--rw metric?          uint32
      +--rw set-metric-type
      |   +--rw metric-type?    identityref
      +--rw set-route-level
      |   +--rw route-level?    identityref
      +--rw set-route-preference?  uint16
      +--rw set-tag?              tag-type
      +--rw set-application-tag?  tag-type
      +--rw bp:bgp-actions
      |   +--rw bp:set-route-origin?bt:bgp-origin-attr-type

```

```

+--rw bp:set-local-pref?    uint32
+--rw bp:set-next-hop?     bgp-next-hop-type
+--rw bp:set-med?          bgp-set-med-type
+--rw bp:set-as-path-prepend
|   +--rw bp:repeat-n?    uint8
+--rw bp:set-community
|   +--rw bp:method?      enumeration
|   +--rw bp:options?
|   +--rw bp:inline
|   |   +--rw bp:communities*  union
|   +--rw bp:reference
|   |   +--rw bp:community-set-ref?
+--rw bp:set-ext-community
|   +--rw bp:method?      enumeration
|   +--rw bp:options?
|   +--rw bp:inline
|   |   +--rw bp:communities*  union
|   +--rw bp:reference
|   |   +--rw bp:ext-community-set-ref?

```

Appendix B. Policy examples

Below we show examples of XML-encoded configuration data using the routing policy and BGP models to illustrate both how policies are defined, and how they can be applied. Note that the XML has been simplified for readability.

The following example shows how prefix-set and tag-set can be defined. The policy condition is to match a prefix-set and a tag-set, and the action is to accept routes that match the condition.

```

<config xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <routing-policy
    xmlns="urn:ietf:params:xml:ns:yang:ietf-routing-policy">

    <defined-sets>
      <prefix-sets>
        <prefix-set>
          <name>prefix-set-A</name>
          <mode>ipv4</mode>
          <prefixes>
            <prefix-list>
              <ip-prefix>192.0.2.0/24</ip-prefix>
              <mask-length-lower>24</mask-length-lower>
              <mask-length-upper>32</mask-length-upper>
            </prefix-list>
            <prefix-list>
              <ip-prefix>198.51.100.0/24</ip-prefix>
            </prefix-list>
          </prefixes>
        </prefix-set>
      </prefix-sets>
    </defined-sets>
  </routing-policy>
</config>

```

```
        <mask-length-lower>24</mask-length-lower>
        <mask-length-upper>32</mask-length-upper>
    </prefix-list>
</prefixes>
</prefix-set>
<prefix-set>
  <name>prefix-set-B</name>
  <mode>ipv6</mode>
  <prefixes>
    <prefix-list>
      <ip-prefix>2001:DB8::/32</ip-prefix>
      <mask-length-lower>32</mask-length-lower>
      <mask-length-upper>64</mask-length-upper>
    </prefix-list>
  </prefixes>
</prefix-set>
</prefix-sets>
<tag-sets>
  <tag-set>
    <name>cust-tag1</name>
    <tag-value>10</tag-value>
  </tag-set>
</tag-sets>
</defined-sets>

<policy-definitions>
  <policy-definition>
    <name>export-tagged-BGP</name>
    <statements>
      <statement>
        <name>term-0</name>
        <conditions>
          <match-prefix-set>
            <prefix-set>prefix-set-A</prefix-set>
          </match-prefix-set>
          <match-tag-set>
            <tag-set>cust-tag1</tag-set>
          </match-tag-set>
        </conditions>
        <actions>
          <policy-result>accept-route</policy-result>
        </actions>
      </statement>
    </statements>
  </policy-definition>
</policy-definitions>

</routing-policy>
```

```
</config>
```

In the following example, all routes in the RIB that have been learned from OSPF advertisements corresponding to OSPF intra-area and inter-area route types should get advertised into ISIS level-2 advertisements.

```
<config xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <routing-policy
    xmlns="urn:ietf:params:xml:ns:yang:ietf-routing-policy">
    <policy-definitions>
      <policy-definition>
        <name>export-all-OSPF-prefixes-into-ISIS-level-2</name>
        <statements>
          <statement>
            <name>term-0</name>
            <conditions>
              <match-route-type>ospf-internal-type</match-route-type>
            </conditions>
            <actions>
              <set-route-level>
                <route-level>isis-level-2</route-level>
              </set-route-level>
              <policy-result>accept-route</policy-result>
            </actions>
          </statement>
        </statements>
      </policy-definition>
    </policy-definitions>
  </routing-policy>
</config>
```

Authors' Addresses

Yingzhen Qu
Futurewei
2330 Central Expressway
Santa Clara CA 95050
USA

Email: yingzhen.qu@futurewei.com

Jeff Tantsura
Microsoft

Email: jefftant.ietf@gmail.com

Acee Lindem
Cisco
301 Midenhall Way
Cary, NC 27513
US

Email: acee@cisco.com

Xufeng Liu
Volta Networks

Email: xufeng.liu.ietf@gmail.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: July 25, 2022

S. Litkowski
Cisco Systems
A. Bashandy
Individual
C. Filsfils
Cisco Systems
P. Francois
INSA Lyon
B. Decraene
Orange
D. Voyer
Bell Canada
January 21, 2022

Topology Independent Fast Reroute using Segment Routing
draft-ietf-rtgwg-segment-routing-ti-lfa-08

Abstract

This document presents Topology Independent Loop-free Alternate Fast Re-route (TI-LFA), aimed at providing protection of node and adjacency segments within the Segment Routing (SR) framework. This Fast Re-route (FRR) behavior builds on proven IP-FRR concepts being LFAs, remote LFAs (RLFA), and remote LFAs with directed forwarding (DLFA). It extends these concepts to provide guaranteed coverage in any two connected network using a link-state IGP. A key aspect of TI-LFA is the FRR path selection approach establishing protection over the expected post-convergence paths from the point of local repair, reducing the operational need to control the tie-breaks among various FRR options.

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 25, 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 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. Acronyms	3
2. Introduction	3
2.1. Conventions used in this document	8
3. Terminology	8
4. Base principle	9
5. Intersecting P-Space and Q-Space with post-convergence paths	9
5.1. Extended P-Space property computation for a resource X, over post-convergence paths	9
5.2. Q-Space property computation for a resource X, over post-convergence paths	10
5.3. Scaling considerations when computing Q-Space	10
6. TI-LFA Repair path	10
6.1. FRR path using a direct neighbor	10
6.2. FRR path using a PQ node	11
6.3. FRR path using a P node and Q node that are adjacent	11
6.4. Connecting distant P and Q nodes along post-convergence paths	11
7. Building TI-LFA repair lists	11
7.1. Link protection	11
7.1.1. The active segment is a node segment	12
7.1.2. The active segment is an adjacency segment	12
7.2. Dataplane specific considerations	13
7.2.1. MPLS dataplane considerations	13
7.2.2. SRv6 dataplane considerations	13
8. TI-LFA and SR algorithms	14
9. Usage of Adjacency segments in the repair list	15
10. Analysis based on real network topologies	15
11. Security Considerations	20
12. IANA Considerations	20
13. Contributors	20
14. Acknowledgments	21

15. References	21
15.1. Normative References	21
15.2. Informative References	22
Authors' Addresses	23

1. Acronyms

- o DLFA: Remote LFA with Directed forwarding.
- o FRR: Fast Re-route.
- o IGP: Interior Gateway Protocol.
- o LFA: Loop-Free Alternate.
- o LSDB: Link State DataBase.
- o PLR: Point of Local Repair.
- o RL: Repair list.
- o RLFA: Remote LFA.
- o RSPT: Reverse Shortest Path Tree.
- o SID: Segment Identifier.
- o SLA: Service Level Agreement.
- o SPF: Shortest Path First.
- o SPT: Shortest Path Tree.
- o SR: Segment Routing.
- o SRGB: Segment Routing Global Block.
- o SRLG: Shared Risk Link Group.
- o TI-LFA: Topology Independant LFA.

2. Introduction

Segment Routing aims at supporting services with tight SLA guarantees [RFC8402]. By relying on SR this document provides a local repair mechanism for standard link-state IGP shortest path capable of restoring end-to-end connectivity in the case of a sudden directly connected failure of a network component. Non-SR mechanisms for

local repair are beyond the scope of this document. Non-local failures are addressed in a separate document [I-D.bashandy-rtgwg-segment-routing-uloop].

The term topology independent (TI) refers to the ability to provide a loop free backup path irrespective of the topologies used in the network. This provides a major improvement compared to LFA [RFC5286] and remote LFA [RFC7490] which cannot provide a complete protection coverage in some topologies as described in [RFC6571].

For each destination in the network, TI-LFA pre-installs a backup forwarding entry for each protected destination ready to be activated upon detection of the failure of a link used to reach the destination. TI-LFA provides protection in the event of any one of the following: single link failure, single node failure, or single SRLG failure. In link failure mode, the destination is protected assuming the failure of the link. In node protection mode, the destination is protected assuming that the neighbor connected to the primary link has failed. In SRLG protecting mode, the destination is protected assuming that a configured set of links sharing fate with the primary link has failed (e.g. a linecard or a set of links sharing a common transmission pipe).

Protection techniques outlined in this document are limited to protecting links, nodes, and SRLGs that are within a link-state IGP area. Protecting domain exit routers and/or links attached to another routing domains are beyond the scope of this document

Thanks to SR, TI-LFA does not require the establishment of TLDP sessions with remote nodes in order to take advantage of the applicability of remote LFAs (RLFA) [RFC7490][RFC7916] or remote LFAs with directed forwarding (DLFA) [RFC5714]. All the Segment Identifiers (SIDs) are available in the link state database (LSDB) of the IGP. As a result, preferring LFAs over RLFA or DLFA, as well as minimizing the number of RLFA or DLFA repair nodes is not required anymore.

Thanks to SR, there is no need to create state in the network in order to enforce an explicit FRR path. This relieves the nodes themselves from having to maintain extra state, and it relieves the operator from having to deploy an extra protocol or extra protocol sessions just to enhance the protection coverage.

[RFC7916] raised several operational considerations when using LFA or remote LFA. [RFC7916] Section 3 presents a case where a high bandwidth link between two core routers is protected through a PE router connected with low bandwidth links. In such a case, congestion may happen when the FRR backup path is activated.

[RFC7916] introduces a local policy framework to let the operator tuning manually the best alternate election based on its own requirements.

From a network capacity planning point of view, it is often assumed that if a link L fails on a particular node X, the bandwidth consumed on L will be spread over some of the remaining links of X. The remaining links to be used are determined by the IGP routing considering that the link L has failed (we assume that the traffic uses the post-convergence path starting from the node X). In Figure 1, we consider a network with all metrics equal to 1 except the metrics on links used by PE1, PE2 and PE3 which are 1000. An easy network capacity planning method is to consider that if the link L (X-B) fails, the traffic actually flowing through L will be spread over the remaining links of X (X-H, X-D, X-A). Considering the IGP metrics, only X-H and X-D can be used in reality to carry the traffic flowing through the link L. As a consequence, the bandwidth of links X-H and X-D is sized according to this rule. We should observe that this capacity planning policy works, however it is not fully accurate.

In Figure 1, considering that the source of traffic is only from PE1 and PE4, when the link L fails, depending on the convergence speed of the nodes, X may reroute its forwarding entries to the remote PEs onto X-H or X-D; however in a similar timeframe, PE1 will also reroute a subset of its traffic (the subset destined to PE2) out of its nominal path reducing the quantity of traffic received by X. The capacity planning rule presented previously has the drawback of oversizing the network, however it allows to prevent any transient congestion (when for example X reroutes traffic before PE1 does).

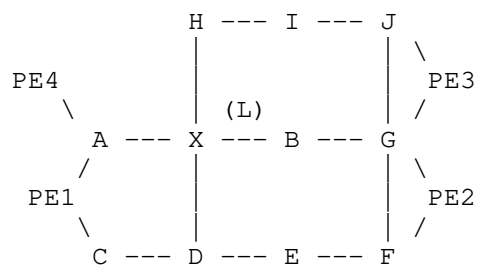


Figure 1

Based on this assumption, in order to facilitate the operation of FRR, and limit the implementation of local FRR policies, it looks interesting to steer the traffic onto the post-convergence path from

the PLR point of view during the FRR phase. In our example, when link L fails, X switches the traffic destined to PE3 and PE2 on the post-convergence paths. This is perfectly inline with the capacity planning rule that was presented before and also inline with the fact X may converge before PE1 (or any other upstream router) and may spread the X-B traffic onto the post-convergence paths rooted at X.

It should be noted, that some networks may have a different capacity planning rule, leading to an allocation of less bandwidth on X-H and X-D links. In such a case, using the post-convergence paths rooted at X during FRR may introduce some congestion on X-H and X-D links. However it is important to note, that a transient congestion may possibly happen, even without FRR activated, for instance when X converges before the upstream routers. Operators are still free to use the policy framework defined in [RFC7916] if the usage of the post-convergence paths rooted at the PLR is not suitable.

Readers should be aware that FRR protection is pre-computing a backup path to protect against a particular type of failure (link, node, SRLG). When using the post-convergence path as FRR backup path, the computed post-convergence path is the one considering the failure we are protecting against. This means that FRR is using an expected post-convergence path, and this expected post-convergence path may be actually different from the post-convergence path used if the failure that happened is different from the failure FRR was protecting against. As an example, if the operator has implemented a protection against a node failure, the expected post-convergence path used during FRR will be the one considering that the node has failed. However, even if a single link is failing or a set of links is failing (instead of the full node), the node-protecting post-convergence path will be used. The consequence is that the path used during FRR is not optimal with respect to the failure that has actually occurred.

Another consideration to take into account is: while using the expected post-convergence path for SR traffic using node segments only (for instance, PE to PE traffic using shortest path) has some advantages, these advantages reduce when SR policies ([I-D.ietf-spring-segment-routing-policy]) are involved. A segment-list used in an SR policy is computed to obey a set of path constraints defined locally at the head-end or centrally in a controller. TI-LFA cannot be aware of such path constraints and there is no reason to expect the TI-LFA backup path protecting one the segments in that segment list to obey those constraints. When SR policies are used and the operator wants to have a backup path which still follows the policy requirements, this backup path should be computed as part of the SR policy in the ingress node (or central controller) and the SR policy should not rely on local protection.

Another option could be to use FlexAlgo ([I-D.ietf-lsr-flex-algo]) to express the set of constraints and use a single node segment associated with a FlexAlgo to reach the destination. When using a node segment associated with a FlexAlgo, TI-LFA keeps providing an optimal backup by applying the appropriate set of constraints. The relationship between TI-LFA and the SR-algorithm is detailed in Section 8.

Thanks to SR and the combination of Adjacency segments and Node segments, the expression of the expected post-convergence path rooted at the PLR is facilitated and does not create any additional state on intermediate nodes. The easiest way to express the expected post-convergence path in a loop-free manner is to encode it as a list of adjacency segments. However, this may create a long SID list that some hardware may not be able to push. One of the challenges of TI-LFA is to encode the expected post-convergence path by combining adjacency segments and node segments. Each implementation will be free to have its own SID list compression optimization algorithm. This document details the basic concepts that could be used to build the SR backup path as well as the associated dataplane procedures.

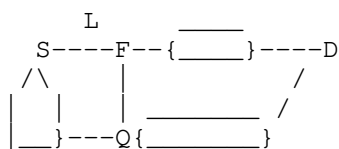


Figure 2: TI-LFA Protection

We use Figure 2 to illustrate the TI-LFA approach.

The Point of Local Repair (PLR), S, needs to find a node Q (a repair node) that is capable of safely forwarding the traffic to a destination D affected by the failure of the protected link L, a set of links including L (SRLG), or the node F itself. The PLR also needs to find a way to reach Q without being affected by the convergence state of the nodes over the paths it wants to use to reach Q: the PLR needs a loop-free path to reach Q.

Section 3 defines the main notations used in the document. They are in line with [RFC5714].

Section 5 suggests to compute the P-Space and Q-Space properties defined in Section 3, for the specific case of nodes lying over the post-convergence paths towards the protected destinations.

Using the properties defined in Section 5, Section 6 describes how to compute protection lists that encode a loop-free post-convergence path towards the destination.

Section 7 defines the segment operations to be applied by the PLR to ensure consistency with the forwarding state of the repair node.

By applying the algorithms specified in this document to actual service providers and large enterprise networks, we provide real life measurements for the number of SIDs used by repair paths. Section 10 summarizes these measurements.

2.1. Conventions used in this document

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. Terminology

We define the main notations used in this document as the following.

We refer to "old" and "new" topologies as the LSDB state before and after the considered failure.

SPT_old(R) is the Shortest Path Tree rooted at node R in the initial state of the network.

SPT_new(R, X) is the Shortest Path Tree rooted at node R in the state of the network after the resource X has failed.

PLR stands for "Point of Local Repair". It is the router that applies fast traffic restoration after detecting failure in a directly attached link, set of links, and/or node.

Similar to [RFC7490], we use the concept of P-Space and Q-Space for TI-LFA.

The P-space $P(R, X)$ of a router R w.r.t. a resource X (e.g. a link S-F, a node F, or a SRLG) is the set of routers reachable from R using the pre-convergence shortest paths without any of those paths (including equal-cost path splits) transiting through X.

Consider the set of neighbors of a router R and a resource X. Exclude from that set of neighbors that are reachable from R using X. The Extended P-Space $P'(R, X)$ of a node R w.r.t. a resource X is the

union of the P-spaces of the neighbors in that reduced set of neighbors w.r.t. the resource X.

The Q-space $Q(R,X)$ of a router R w.r.t. a resource X is the set of routers from which R can be reached without any path (including equal-cost path splits) transiting through X.

A symmetric network is a network such that the IGP metric of each link is the same in both directions of the link.

4. Base principle

The basic algorithm to compute the repair path is to pre-compute $SPT_new(R,X)$ and for each destination, encode the repair path as a loop-free SID list. One way to provide a loop-free SID list is to use adjacency SIDs only. However, this approach may create very long SID lists that hardware may not be able to handle due to MSD (Maximum SID Depth) limitations.

An implementation is free to use any local optimization to provide smaller SID lists by combining Node SIDs and Adjacency SIDs. In addition, the usage of Node-SIDs allow to maximize ECMPs over the backup path. These optimizations are out of scope of this document, however the subsequent sections provide some guidance on how to leverage P-Spaces and Q-Spaces to optimize the size of the SID list.

5. Intersecting P-Space and Q-Space with post-convergence paths

One of the challenges of defining an SR path following the expected post-convergence path is to reduce the size of the segment list. In order to reduce this segment list, an implementation MAY determine the P-Space/Extended P-Space and Q-Space properties (defined in [RFC7490]) of the nodes along the expected post-convergence path from the PLR to the protected destination and compute an SR-based explicit path from P to Q when they are not adjacent. Such properties will be used in Section 6 to compute the TI-LFA repair list.

5.1. Extended P-Space property computation for a resource X, over post-convergence paths

We want to determine which nodes on the post-convergence path from the PLR R to the destination D are in the extended P-space of R w.r.t. resource X (X can be a link or a set of links adjacent to the PLR, or a neighbor node of the PLR).

This can be found by intersecting the set of nodes belonging to the post-convergence path from R to D, assuming the failure of X, with $P'(R, X)$.

5.2. Q-Space property computation for a resource X, over post-convergence paths

We want to determine which nodes on the post-convergence path from the PLR R to the destination D are in the Q-Space of destination D w.r.t. resource X (X can be a link or a set of links adjacent to the PLR, or a neighbor node of the PLR).

This can be found by intersecting the set of nodes belonging to the post-convergence path from R to D, assuming the failure of X, with $Q(D, X)$.

5.3. Scaling considerations when computing Q-Space

[RFC7490] raises scaling concerns about computing a Q-Space per destination. Similar concerns may affect TI-LFA computation if an implementation tries to compute a reverse Shortest Path Tree ([RFC7490]) for every destination in the network to determine the Q-Space. It will be up to each implementation to determine the good tradeoff between scaling and accuracy of the optimization.

6. TI-LFA Repair path

The TI-LFA repair path (RP) consists of an outgoing interface and a list of segments (repair list (RL)) to insert on the SR header. The repair list encodes the explicit post-convergence path to the destination, which avoids the protected resource X and, at the same time, is guaranteed to be loop-free irrespective of the state of FIBs along the nodes belonging to the explicit path. Thus, there is no need for any co-ordination or message exchange between the PLR and any other router in the network.

The TI-LFA repair path is found by intersecting $P(S, X)$ and $Q(D, X)$ with the post-convergence path to D and computing the explicit SR-based path $EP(P, Q)$ from P to Q when these nodes are not adjacent along the post convergence path. The TI-LFA repair list is expressed generally as $(Node_SID(P), EP(P, Q))$.

Most often, the TI-LFA repair list has a simpler form, as described in the following sections. Section 10 provides statistics for the number of SIDs in the explicit path to protect against various failures.

6.1. FRR path using a direct neighbor

When a direct neighbor is in $P(S, X)$ and $Q(D, x)$ and on the post-convergence path, the outgoing interface is set to that neighbor and the repair segment list SHOULD be empty.

This is comparable to a post-convergence LFA FRR repair.

6.2. FRR path using a PQ node

When a remote node R is in $P(S,X)$ and $Q(D,x)$ and on the post-convergence path, the repair list MUST be made of a single node segment to R and the outgoing interface SHOULD be set to the outgoing interface used to reach R.

This is comparable to a post-convergence RLFA repair tunnel.

6.3. FRR path using a P node and Q node that are adjacent

When a node P is in $P(S,X)$ and a node Q is in $Q(D,x)$ and both are on the post-convergence path and both are adjacent to each other, the repair list SHOULD be made of two segments: A node segment to P (to be processed first), followed by an adjacency segment from P to Q.

This is comparable to a post-convergence DLFA (LFA with directed forwarding) repair tunnel.

6.4. Connecting distant P and Q nodes along post-convergence paths

In some cases, there is no adjacent P and Q node along the post-convergence path. As mentioned in Section 4, a list of adjacency SIDs can be used to encode the path between P and Q. However, the PLR can perform additional computations to compute a list of segments that represent a loop-free path from P to Q. How these computations are done is out of scope of this document and is left to implementation.

7. Building TI-LFA repair lists

The following sections describe how to build the repair lists using the terminology defined in [RFC8402]. The procedures described in Section 7.1 are equally applicable to both SR-MPLS and SRv6 dataplane, while the dataplane-specific considerations are described in Section 7.2.

7.1. Link protection

In this section, we explain how a protecting router S processes the active segment of a packet upon the failure of its primary outgoing interface for the packet, S-F.

7.1.1. The active segment is a node segment

The active segment MUST be kept on the SR header unchanged and the repair list MUST be added. The active segment becomes the first segment of the repair list. The way the repair list is added depends on the dataplane used (see Section 7.2).

7.1.2. The active segment is an adjacency segment

We define hereafter the FRR behavior applied by S for any packet received with an active adjacency segment S-F for which protection was enabled. As protection has been enabled for the segment S-F and signaled in the IGP (for instance using protocol extensions from [RFC8667] and [RFC8665]), any SR policy using this segment knows that it may be transiently rerouted out of S-F in case of S-F failure.

The simplest approach for link protection of an adjacency segment S-F is to create a repair list that will carry the traffic to F. To do so, one or more "PUSH" operations are performed. If the repair list, while avoiding S-F, terminates on F, S only pushes segments of the repair list. Otherwise, S pushes a node segment of F, followed by the segments of the repair list. For details on the "NEXT" and "PUSH" operations, refer to [RFC8402].

This method which merges back the traffic at the remote end of the adjacency segment has the advantage of keeping as much as possible the traffic on the pre-failure path. As stated in Section 2, when SR policies are involved and a strict compliance of the policy is required, an end-to-end protection should be preferred over a local repair mechanism. However, this method may not provide the expected post-convergence path to the final destination as the expected post-convergence path may not go through F. Another method requires to look to the next segment in the segment list.

We distinguish the case where this active segment is followed by another adjacency segment from the case where it is followed by a node segment.

7.1.2.1. Protecting [Adjacency, Adjacency] segment lists

If the next segment in the list is an Adjacency segment, then the packet has to be conveyed to F.

To do so, S MUST apply a "NEXT" operation on Adj(S-F) and then one or more "PUSH" operations. If the repair list, while avoiding S-F, terminates on F, S only pushes the segments of the repair list. Otherwise, S pushes a node segment of F, followed by the segments of

the repair list. For details on the "NEXT" and "PUSH" operations, refer to [RFC8402].

Upon failure of S-F, a packet reaching S with a segment list matching [adj(S-F),adj(F-M),...] will thus leave S with a segment list matching [RL(F),node(F),adj(F-M),...], where RL(F) is the repair path for destination F.

7.1.2.2. Protecting [Adjacency, Node] segment lists

If the next segment in the stack is a node segment, say for node T, the segment list on the packet matches [adj(S-F),node(T),...].

In this case, S MUST apply a "NEXT" operation on the Adjacency segment related to S-F, followed by a "PUSH" of a repair list redirecting the traffic to a node Q, whose path to node segment T is not affected by the failure.

Upon failure of S-F, packets reaching S with a segment list matching [adj(S-F), node(T), ...], would leave S with a segment list matching [RL(Q),node(T), ...].

7.2. Dataplane specific considerations

7.2.1. MPLS dataplane considerations

MPLS dataplane for Segment Routing is described in [RFC8660].

The following dataplane behaviors apply when creating a repair list using an MPLS dataplane:

1. If the active segment is a node segment that has been signaled with penultimate hop popping and the repair list ends with an adjacency segment terminating on the tail-end of the active segment, then the active segment MUST be popped before pushing the repair list.
2. If the active segment is a node segment but the other conditions in 1. are not met, the active segment MUST be popped then pushed again with a label value computed according to the SRGB of Q, where Q is the endpoint of the repair list. Finally, the repair list MUST be pushed.

7.2.2. SRv6 dataplane considerations

SRv6 dataplane and programming instructions are described respectively in [RFC8754] and [RFC8986].

The TI-LFA path computation algorithm is the same as in the SR-MPLS dataplane. Note however that the Adjacency SIDs are typically globally routed. In such case, there is no need for a preceding Prefix SID and the resulting repair list is likely shorter.

If the traffic is protected at a Transit Node, then an SRv6 SID list is added on the packet to apply the repair list. The addition of the repair list follows the headend behaviors as specified in section 5 of [RFC8986].

If the traffic is protected at an SR Segment Endpoint Node, first the Segment Endpoint packet processing is executed. Then the packet is protected as if its were a transit packet.

8. TI-LFA and SR algorithms

SR allows an operator to bind an algorithm to a prefix SID (as defined in [RFC8402]). The algorithm value dictates how the path to the prefix is computed. The SR default algorithm is known as the "Shortest Path" algorithm. The SR default algorithm allows an operator to override the IGP shortest path by using local policies. When TI-LFA uses Node-SIDs associated with the default algorithm, there is no guarantee that the path will be loop-free as a local policy may have overridden the expected IGP path. As the local policies are defined by the operator, it becomes the responsibility of this operator to ensure that the deployed policies do not affect the TI-LFA deployment. It should be noted that such situation can already happen today with existing mechanisms as remote LFA.

[I-D.ietf-lsr-flex-algo] defines a flexible algorithm (FlexAlgo) framework to be associated with Prefix SIDs. FlexAlgo allows a user to associate a constrained path to a Prefix SID rather than using the regular IGP shortest path. An implementation MAY support TI-LFA to protect Node-SIDs associated to a FlexAlgo. In such a case, rather than computing the expected post-convergence path based on the regular SPF, an implementation SHOULD use the constrained SPF algorithm bound to the FlexAlgo (using the Flex Algo Definition) instead of the regular Dijkstra in all the SPF/rSPF computations that are occurring during the TI-LFA computation. This includes the computation of the P-Space and Q-Space as well as the post-convergence path. An implementation MUST only use Node-SIDs bound to the FlexAlgo and/or Adj-SIDs that are unprotected to build the repair list.

9. Usage of Adjacency segments in the repair list

The repair list of segments computed by TI-LFA may contain one or more adjacency segments. An adjacency segment may be protected or not protected.

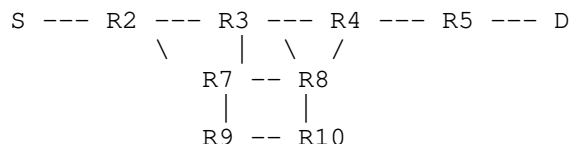


Figure 3

In Figure 3, all the metrics are equal to 1 except R2-R7, R7-R8, R8-R4, R7-R9 which have a metric of 1000. Considering R2 as a PLR to protect against the failure of node R3 for the traffic S->D, the repair list computed by R2 will be [adj(R7-R8), adj(R8-R4)] and the outgoing interface will be to R7. If R3 fails, R2 pushes the repair list onto the incoming packet to D. During the FRR, if R7-R8 fails and if TI-LFA has picked a protected adjacency segment for adj(R7-R8), R7 will push an additional repair list onto the packet following the procedures defined in Section 7.

To avoid the possibility of this double FRR activation, an implementation of TI-LFA MAY pick only non protected adjacency segments when building the repair list. However, this is important to note that FRR in general is intended to protect for a single pre-planned failure. If the failure that happens is worse than expected or multiple failures happen, FRR is not guaranteed to work. In such a case, fast IGP convergence remains important to restore traffic as quickly as possible.

10. Analysis based on real network topologies

This section presents analysis performed on real service provider and large enterprise network topologies. The objective of the analysis is to assess the number of SIDs required in an explicit path when the mechanisms described in this document are used to protect against the failure scenarios within the scope of this document. The number of segments described in this section are applicable to instantiating segment routing over the MPLS forwarding plane.

The measurement below indicate that for link and local SRLG protection, a 1 SID repair path delivers more than 99% coverage. For node protection a 2 SIDs repair path yields 99% coverage.

Table 1 below lists the characteristics of the networks used in our measurements. The number of links refers to the number of "bidirectional" links (not directed edges of the graph). The measurements are carried out as follows:

- o For each network, the algorithms described in this document are applied to protect all prefixes against link, node, and local SRLG failure
- o For each prefix, the number of SIDs used by the repair path is recorded
- o The percentage of number of SIDs are listed in Tables 2A/B, 3A/B, and 4A/B

The measurements listed in the tables indicate that for link and local SRLG protection, 1 SID repair paths are sufficient to protect more than 99% of the prefix in almost all cases. For node protection 2 SIDs repair paths yield 99% coverage.

Network	Nodes	Links	Node-to-Link Ratio	SRLG info?
T1	408	665	1.63	Yes
T2	587	1083	1.84	No
T3	93	401	4.31	Yes
T4	247	393	1.59	Yes
T5	34	96	2.82	Yes
T6	50	78	1.56	No
T7	82	293	3.57	No
T8	35	41	1.17	Yes
T9	177	1371	7.74	Yes

Table 1: Data Set Definition

The rest of this section presents the measurements done on the actual topologies. The convention that we use is as follows

- o 0 SIDs: the calculated repair path starts with a directly connected neighbor that is also a loop free alternate, in which case there is no need to explicitly route the traffic using additional SIDs. This scenario is described in Section 6.1.
- o 1 SIDs: the repair node is a PQ node, in which case only 1 SID is needed to guarantee loop-freeness. This scenario is covered in Section 6.2.
- o 2 or more SIDs: The repair path consists of 2 or more SIDs as described in Section 6.3 and Section 6.4. We do not cover the case for 2 SIDs (Section 6.3) separately because there was no granularity in the result. Also we treat the node-SID+adj-SID and node-SID + node-SID the same because they do not differ from the data plane point of view.

Table 2A and 2B below summarize the measurements on the number of SIDs needed for link protection

Network	0 SIDs	1 SID	2 SIDs	3 SIDs
T1	74.3%	25.3%	0.5%	0.0%
T2	81.1%	18.7%	0.2%	0.0%
T3	95.9%	4.1%	0.1%	0.0%
T4	62.5%	35.7%	1.8%	0.0%
T5	85.7%	14.3%	0.0%	0.0%
T6	81.2%	18.7%	0.0%	0.0%
T7	98.9%	1.1%	0.0%	0.0%
T8	94.1%	5.9%	0.0%	0.0%
T9	98.9%	1.0%	0.0%	0.0%

Table 2A: Link protection (repair size distribution)

Network	0 SIDs	1 SID	2 SIDs	3 SIDs
T1	74.2%	99.5%	99.9%	100.0%
T2	81.1%	99.8%	100.0%	100.0%

T3	95.9%	99.9%	100.0%	100.0%
T4	62.5%	98.2%	100.0%	100.0%
T5	85.7%	100.0%	100.0%	100.0%
T6	81.2%	99.9%	100.0%	100.0%
T7	98,8%	100.0%	100.0%	100.0%
T8	94,1%	100.0%	100.0%	100.0%
T9	98,9%	100.0%	100.0%	100.0%

Table 2B: Link protection repair size cumulative distribution
 Table 3A and 3B summarize the measurements on the number of SIDs needed for local SRLG protection.

Network	0 SIDs	1 SID	2 SIDs	3 SIDs
T1	74.2%	25.3%	0.5%	0.0%
T2	No SRLG Information			
T3	93.6%	6.3%	0.0%	0.0%
T4	62.5%	35.6%	1.8%	0.0%
T5	83.1%	16.8%	0.0%	0.0%
T6	No SRLG Information			
T7	No SRLG Information			
T8	85.2%	14.8%	0.0%	0.0%
T9	98,9%	1.1%	0.0%	0.0%

Table 3A: Local SRLG protection repair size distribution

Network	0 SIDs	1 SID	2 SIDs	3 SIDs
T1	74.2%	99.5%	99.9%	100.0%
T2	No SRLG Information			

T3	93.6%	99.9%	100.0%	0.0%
T4	62.5%	98.2%	100.0%	100.0%
T5	83.1%	100.0%	100.0%	100.0%
T6	No SRLG Information			
T7	No SRLG Information			
T8	85.2%	100.0%	100.0%	100.0%
T9	98.9%	100.0%	100.0%	100.0%

Table 3B: Local SRLG protection repair size Cumulative distribution
The remaining two tables summarize the measurements on the number of SIDs needed for node protection.

Network	0 SIDs	1 SID	2 SIDs	3 SIDs	4 SIDs
T1	49.8%	47.9%	2.1%	0.1%	0.0%
T2	36.5%	59.6%	3.6%	0.2%	0.0%
T3	73.3%	25.6%	1.1%	0.0%	0.0%
T4	36.1%	57.3%	6.3%	0.2%	0.0%
T5	73.2%	26.8%	0%	0%	0%
T6	78.3%	21.3%	0.3%	0%	0%
T7	66.1%	32.8%	1.1%	0%	0%
T8	59.7%	40.2%	0%	0%	0%
T9	98.9%	1.0%	0%	0%	0%

Table 4A: Node protection (repair size distribution)

Network	0 SIDs	1 SID	2 SIDs	3 SIDs	4 SIDs
T1	49.7%	97.6%	99.8%	99.9%	100%
T2	36.5%	96.1%	99.7%	99.9%	100%

T3	73.3%	98.9%	99.9%	100.0%	100%
T4	36.1%	93.4%	99.8%	99.9%	100%
T5	73.2%	100.0%	100.0%	100.0%	100%
T6	78.4%	99.7%	100.0%	100.0%	100%
T7	66.1%	98.9%	100.0%	100.0%	100%
T8	59.7%	100.0%	100.0%	100.0%	100%
T9	98.9%	100.0%	100.0%	100.0%	100%

Table 4B: Node protection (repair size cumulative distribution)

11. Security Considerations

The techniques described in this document are internal functionalities to a router that result in the ability to guarantee an upper bound on the time taken to restore traffic flow upon the failure of a directly connected link or node. As these techniques steer traffic to the post-convergence path as quickly as possible, this serves to minimize the disruption associated with a local failure which can be seen as a modest security enhancement. The protection mechanisms does not protect external destinations, but rather provides quick restoration for destination that are internal to a routing domain.

Security considerations described in [RFC5286] and [RFC7490] apply to this document. Similarly, as the solution described in the document is based on Segment Routing technology, reader should be aware of the security considerations related to this technology ([RFC8402]) and its dataplane instantiations ([RFC8660], [RFC8754] and [RFC8986]). However, this document does not introduce additional security concern.

12. IANA Considerations

No requirements for IANA

13. Contributors

In addition to the authors listed on the front page, the following co-authors have also contributed to this document:

Francois Clad, Cisco Systems

Pablo Camarillo, Cisco Systems

14. Acknowledgments

We would like to thank Les Ginsberg, Stewart Bryant, Alexander Vainsthein, Chris Bowers, Shraddha Hedge for their valuable comments.

15. References

15.1. 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>>.
- [RFC7916] Litkowski, S., Ed., Decraene, B., Filsfils, C., Raza, K., Horneffer, M., and P. Sarkar, "Operational Management of Loop-Free Alternates", RFC 7916, DOI 10.17487/RFC7916, July 2016, <<https://www.rfc-editor.org/info/rfc7916>>.
- [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>>.
- [RFC8660] Bashandy, A., Ed., Filsfils, C., Ed., Previdi, S., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing with the MPLS Data Plane", RFC 8660, DOI 10.17487/RFC8660, December 2019, <<https://www.rfc-editor.org/info/rfc8660>>.
- [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>>.
- [RFC8986] Filsfils, C., Ed., Camarillo, P., Ed., Leddy, J., Voyer, D., Matsushima, S., and Z. Li, "Segment Routing over IPv6 (SRv6) Network Programming", RFC 8986, DOI 10.17487/RFC8986, February 2021, <<https://www.rfc-editor.org/info/rfc8986>>.

15.2. Informative References

- [I-D.bashandy-rtgwg-segment-routing-uloop]
Bashandy, A., Filsfils, C., Litkowski, S., Decraene, B., Francois, P., and P. Psenak, "Loop avoidance using Segment Routing", draft-bashandy-rtgwg-segment-routing-uloop-12 (work in progress), December 2021.
- [I-D.ietf-lsr-flex-algo]
Psenak, P., Hegde, S., Filsfils, C., Talaulikar, K., and A. Gulko, "IGP Flexible Algorithm", draft-ietf-lsr-flex-algo-18 (work in progress), October 2021.
- [I-D.ietf-spring-segment-routing-policy]
Filsfils, C., Talaulikar, K., Voyer, D., Bogdanov, A., and P. Mattes, "Segment Routing Policy Architecture", draft-ietf-spring-segment-routing-policy-14 (work in progress), October 2021.
- [RFC5286] Atlas, A., Ed. and A. Zinin, Ed., "Basic Specification for IP Fast Reroute: Loop-Free Alternates", RFC 5286, DOI 10.17487/RFC5286, September 2008, <<https://www.rfc-editor.org/info/rfc5286>>.
- [RFC5714] Shand, M. and S. Bryant, "IP Fast Reroute Framework", RFC 5714, DOI 10.17487/RFC5714, January 2010, <<https://www.rfc-editor.org/info/rfc5714>>.
- [RFC6571] Filsfils, C., Ed., Francois, P., Ed., Shand, M., Decraene, B., Uttaro, J., Leymann, N., and M. Horneffer, "Loop-Free Alternate (LFA) Applicability in Service Provider (SP) Networks", RFC 6571, DOI 10.17487/RFC6571, June 2012, <<https://www.rfc-editor.org/info/rfc6571>>.
- [RFC7490] Bryant, S., Filsfils, C., Previdi, S., Shand, M., and N. So, "Remote Loop-Free Alternate (LFA) Fast Reroute (FRR)", RFC 7490, DOI 10.17487/RFC7490, April 2015, <<https://www.rfc-editor.org/info/rfc7490>>.
- [RFC8665] Psenak, P., Ed., Previdi, S., Ed., Filsfils, C., Gredler, H., Shakir, R., Henderickx, W., and J. Tantsura, "OSPF Extensions for Segment Routing", RFC 8665, DOI 10.17487/RFC8665, December 2019, <<https://www.rfc-editor.org/info/rfc8665>>.

[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>>.

Authors' Addresses

Stephane Litkowski
Cisco Systems
France

Email: slitkows@cisco.com

Ahmed Bashandy
Individual

Email: abashandy.ietf@gmail.com

Clarence Filsfils
Cisco Systems
Brussels
Belgium

Email: cfilsfil@cisco.com

Pierre Francois
INSA Lyon

Email: pierre.francois@insa-lyon.fr

Bruno Decraene
Orange
Issy-les-Moulineaux
France

Email: bruno.decraene@orange.com

Daniel Voyer
Bell Canada
Canada

Email: daniel.voyer@bell.ca

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: 10 November 2022

A. Lindem
Cisco Systems
Y. Qu
Futurewei
9 May 2022

RIB Extension YANG Data Model
draft-ietf-rtgwg-yang-rib-extend-11

Abstract

A Routing Information Base (RIB) is a list of routes and their corresponding administrative data and operational state.

RFC 8349 defines the basic building blocks for RIB, and this model augments it to support multiple next-hops (aka, paths) for each route as well as additional attributes.

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 10 November 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	2
2. Terminology and Notation	3
2.1. Tree Diagrams	4
2.2. Prefixes in Data Node Names	4
3. Design of the Model	4
3.1. Tags and Preference	4
3.2. Repair Path	5
4. RIB Model Tree	6
5. RIB Extension YANG Model	6
6. Security Considerations	13
7. IANA Considerations	15
8. References	15
8.1. Normative References	15
8.2. Informative References	17
Appendix A. Combined Tree Diagram	17
Appendix B. ietf-rib-extension.yang example	20
Appendix C. Acknowledgments	25
Authors' Addresses	25

1. Introduction

This document defines a YANG [RFC7950] data model which extends the RIBs defined in ietf-routing YANG module [RFC8349] with more route attributes.

A RIB is a collection of routes with attributes controlled and manipulated by control-plane protocols. Each RIB contains only routes of one address family [RFC8349]. Within a protocol, routes are selected based on the metrics in use by that protocol, and the protocol installs the routes to RIB. RIB selects the preferred routes by comparing the route-preference (aka, administrative distance) of the associated protocol.

The module defined in this document extends the RIBs to support more route attributes, such as multiple next-hops, route metrics, and administrative tags.

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

2. Terminology and Notation

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]:

- * configuration
- * system state
- * operational state

The following terms are defined in [RFC7950]:

- * action
- * augment
- * container
- * container with presence
- * data model
- * data node
- * leaf
- * list
- * mandatory node
- * module
- * schema tree
- * RPC (Remote Procedure Call) operation

The following terms are defined in [RFC8349] Section 5.2:

- * RIB

2.1. Tree Diagrams

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

2.2. Prefixes in Data Node Names

In this document, names of data nodes, actions, and other data model objects are often used without a prefix, as long as it is clear from the context in which YANG module each name is defined. Otherwise, names are prefixed using the standard prefix associated with the corresponding YANG module, as shown in Table 1.

Prefix	YANG module	Reference
if	ietf-interfaces	[RFC8343]
rt	ietf-routing	[RFC8349]
v4ur	ietf-ipv4-unicast-routing	[RFC8349]
v6ur	ietf-ipv6-unicast-routing	[RFC8349]
inet	ietf-inet-types	[RFC6991]

Table 1: Prefixes and Corresponding YANG Modules

3. Design of the Model

The YANG module defined in this document augments the ietf-routing YANG model defined in [RFC8349], which provides a basis for routing system data model development. Together with YANG modules defined in [RFC8349], a generic RIB YANG model is defined to implement and monitor a RIB.

The models in [RFC8349] also define the basic configuration and operational state for both IPv4 and IPv6 static routes. This document provides augmentations for static routes to support multiple next-hops and more next-hop attributes.

3.1. Tags and Preference

Individual route tags are supported at both the route and next-hop level. A preference per next-hop is also supported for selection of the most preferred reachable static route.

The following tree snapshot shows tag and preference which augment static IPv4 unicast routes and IPv6 unicast routes next-hop.

```

augment /rt:routing/rt:control-plane-protocols
  /rt:control-plane-protocol/rt:static-routes/v4ur:ipv4
  /v4ur:route/v4ur:next-hop/v4ur:next-hop-options
  /v4ur:simple-next-hop:
    +---rw preference?   uint32
    +---rw tag?          uint32
augment /rt:routing/rt:control-plane-protocols
  /rt:control-plane-protocol/rt:static-routes/v4ur:ipv4
  /v4ur:route/v4ur:next-hop/v4ur:next-hop-options
  /v4ur:next-hop-list/v4ur:next-hop-list/v4ur:next-hop:
    +---rw preference?   uint32
    +---rw tag?          uint32
augment /rt:routing/rt:control-plane-protocols
  /rt:control-plane-protocol/rt:static-routes/v6ur:ipv6
  /v6ur:route/v6ur:next-hop/v6ur:next-hop-options
  /v6ur:simple-next-hop:
    +---rw preference?   uint32
    +---rw tag?          uint32
augment /rt:routing/rt:control-plane-protocols
  /rt:control-plane-protocol/rt:static-routes/v6ur:ipv6
  /v6ur:route/v6ur:next-hop/v6ur:next-hop-options
  /v6ur:next-hop-list/v6ur:next-hop-list/v6ur:next-hop:
    +---rw preference?   uint32
    +---rw tag?          uint32

```

3.2. Repair Path

The IP Fast Reroute (IPFRR) pre-computes repair paths by routing protocols [RFC5714], and the repair paths are installed in the RIB.

Each route in the RIB is augmented with repair paths if available, and is shown in the following tree snapshot.

```

augment /rt:routing/rt:ribs/rt:rib/rt:routes/rt:route
  /rt:next-hop/rt:next-hop-options/rt:simple-next-hop:
  +--ro repair-path
    +--ro outgoing-interface?   if:interface-state-ref
    +--ro next-hop-address?     inet:ip-address
    +--ro metric?               uint32
augment /rt:routing/rt:ribs/rt:rib/rt:routes/rt:route
  /rt:next-hop/rt:next-hop-options/rt:next-hop-list
  /rt:next-hop-list/rt:next-hop:
  +--ro repair-path
    +--ro outgoing-interface?   if:interface-state-ref
    +--ro next-hop-address?     inet:ip-address
    +--ro metric?               uint32

```

4. RIB Model Tree

The ietf-routing.yang tree with the augmentations herein is included in Appendix A. The meaning of the symbols can be found in [RFC8340].

5. RIB Extension YANG Model

```

<CODE BEGINS> file "ietf-rib-extension@2021-10-17.yang"
module ietf-rib-extension {
  yang-version "1.1";
  namespace "urn:ietf:params:xml:ns:yang:ietf-rib-extension";

  prefix rib-ext;

  import ietf-inet-types {
    prefix "inet";
    reference "RFC 6991: Common YANG Data Types";
  }

  import ietf-interfaces {
    prefix "if";
    reference "RFC 8343: A YANG Data Model for Interface
              Management (NMDA Version)";
  }

  import ietf-routing {
    prefix "rt";
    reference "RFC 8349: A YANG Data Model for Routing
              Management (NMDA Version)";
  }

  import ietf-ipv4-unicast-routing {
    prefix "v4ur";
    reference "RFC 8349: A YANG Data Model for Routing

```

```

        Management (NMDA Version)";
    }

    import ietf-ipv6-unicast-routing {
        prefix "v6ur";
        reference "RFC 8349: A YANG Data Model for Routing
            Management (NMDA Version)";
    }

    organization
        "IETF RTGWG - Routing Working Group";

    contact
        "WG Web:  <http://datatracker.ietf.org/group/rtgwg/>
        WG List:  <mailto:rtgwg@ietf.org>

        Author:   Acee Lindem
                  <mailto:acee@cisco.com>
        Author:   Yingzhen Qu
                  <mailto:yingzhen.qu@futurewei.com>";

    description
        "This document defines a YANG data model which extends
        the RIBs defined in ietf-routing YANG module with more
        route attributes.

        This YANG model conforms to the Network Management
        Datastore Architecture (NDMA) as described in RFC 8342.

        Copyright (c) 2021 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 2021-10-17 {
        description
            "Initial Version";
        reference
            "RFC XXXX: A YANG Data Model for RIB Extensions.";
    }
}
```

```
/* Groupings */
grouping rib-statistics {
  description
    "Statistics grouping used for RIB augmentation.";
  container statistics {
    config false;
    description
      "Container for RIB statistics.";
    leaf total-routes {
      type uint32;
      description
        "Total routes in the RIB";
    }
    leaf total-active-routes {
      type uint32;
      description
        "Total active routes in the RIB. An active route is
        preferred over other routes to the same destination
        prefix.";
    }
    leaf total-route-memory {
      type uint64;
      units "bytes";
      description
        "Total memory for all routes in the RIB.";
    }
  }
  list protocol-statistics {
    description "RIB statistics per protocol.";
    leaf protocol {
      type identityref {
        base rt:routing-protocol;
      }
      description "Routing protocol.";
    }
    leaf routes {
      type uint32;
      description
        "Total routes for protocol in the RIB.";
    }
    leaf active-routes {
      type uint32;
      description
        "Total active routes for protocol in the RIB. An active
        route is preferred over other routes to the same
        destination prefix.";
    }
    leaf route-memory {
      type uint64;
    }
  }
}
```

```
        units "bytes";
        description
            "Total memory for all routes for protocol in the RIB.";
    }
}
}

grouping next-hop {
    description
        "Next-hop grouping";
    leaf interface {
        type if:interface-ref;
        description
            "Outgoing interface";
    }
    leaf address {
        type inet:ip-address;
        description
            "IPv4 or IPv6 Address of the next-hop.";
    }
}

grouping attributes {
    description
        "Common attributes applicable to all routes.";
    leaf metric {
        type uint32;
        description
            "The metric is a numeric value indicating the cost
            of the route from the perspective of the routing
            protocol installing the route. In general, routes with
            a lower metric installed by the same routing protocol
            are lower cost to reach and are preferable to routes
            with a higher metric. However, metrics from different
            routing protocols are not directly comparable.";
    }
    leaf-list tag {
        type uint32;
        description
            "A tag is a 32-bit opaque value associated with the
            route that can be used for policy decisions such as
            advertisement and filtering of the route.";
    }
    leaf application-tag {
        type uint32;
        description
            "The application-specific tag is an additional tag that
```

```
        can be used by applications that require semantics and/or
        policy different from that of the tag. For example,
        the tag is usually automatically advertised in OSPF
        AS-External Link State Advertisements (LSAs) while this
        application-specific tag is not advertised implicitly.";
    }
}
grouping repair-path {
  description
    "Grouping for IP Fast Reroute repair path.";
  container repair-path {
    description
      "IP Fast Reroute next-hop repair path.";
    leaf outgoing-interface {
      type if:interface-state-ref;
      description
        "Name of the outgoing interface.";
    }
    leaf next-hop-address {
      type inet:ip-address;
      description
        "IP address of the next hop.";
    }
    leaf metric {
      type uint32;
      description
        "The metric for the repair path. While the IP Fast
        Reroute re-route repair is local and the metric is
        not advertised externally, the metric for repair path
        is useful for troubleshooting purposes.";
    }
    reference
      "RFC 5714: IP Fast Reroute Framework.";
  }
}

augment "/rt:routing/rt:control-plane-protocols/"
+ "rt:control-plane-protocol/rt:static-routes/v4ur:ipv4/"
+ "v4ur:route/v4ur:next-hop/v4ur:next-hop-options/"
+ "v4ur:simple-next-hop"
{
  description
    "Augment 'simple-next-hop' case in IPv4 unicast route.";
  leaf preference {
    type uint32;
    default "1";
    description
      "The preference is used to select among multiple static
```



```
        routes. Routes with a lower preference next-hop are
        preferred and equal preference routes result in
        Equal-Cost-Multi-Path (ECMP) static routes.";
    }
    leaf tag {
        type uint32;
        default "0";
        description
            "The tag is a 32-bit opaque value associated with the
            route that can be used for policy decisions such as
            advertisement and filtering of the route.";
    }
}

augment "/rt:routing/rt:control-plane-protocols/"
+ "rt:control-plane-protocol/rt:static-routes/v4ur:ipv4/"
+ "v4ur:route/v4ur:next-hop/v4ur:next-hop-options/"
+ "v4ur:next-hop-list/v4ur:next-hop-list/v4ur:next-hop"
{
    description
        "Augment static route configuration 'next-hop-list'.";

    leaf preference {
        type uint32;
        default "1";
        description
            "The preference is used to select among multiple static
            routes. Routes with a lower preference next-hop are
            preferred and equal preference routes result in
            Equal-Cost-Multi-Path (ECMP) static routes.";
    }

    leaf tag {
        type uint32;
        default "0";
        description
            "The tag is a 32-bit opaque value associated with the
            route that can be used for policy decisions such as
            advertisement and filtering of the route.";
    }
}

augment "/rt:routing/rt:control-plane-protocols/"
+ "rt:control-plane-protocol/rt:static-routes/v6ur:ipv6/"
+ "v6ur:route/v6ur:next-hop/v6ur:next-hop-options/"
+ "v6ur:simple-next-hop"
{
    description
        "Augment 'simple-next-hop' case in IPv6 unicast route.";
```

```
    leaf preference {
      type uint32;
      default "1";
      description
        "The preference is used to select among multiple static
        routes. Routes with a lower preference next-hop are
        preferred and equal preference routes result in
        Equal-Cost-Multi-Path (ECMP) static routes.";
    }
    leaf tag {
      type uint32;
      default "0";
      description
        "The tag is a 32-bit opaque value associated with the
        route that can be used for policy decisions such as
        advertisement and filtering of the route.";
    }
  }
}

augment "/rt:routing/rt:control-plane-protocols/"
+ "rt:control-plane-protocol/rt:static-routes/v6ur:ipv6/"
+ "v6ur:route/v6ur:next-hop/v6ur:next-hop-options/"
+ "v6ur:next-hop-list/v6ur:next-hop-list/v6ur:next-hop"
{
  description
    "Augment static route configuration 'next-hop-list'.";

  leaf preference {
    type uint32;
    default "1";
    description
      "The preference is used to select among multiple static
      routes. Routes with a lower preference next-hop are
      preferred and equal preference routes result in
      Equal-Cost-Multi-Path (ECMP) static routes.";
  }
  leaf tag {
    type uint32;
    default "0";
    description
      "The tag is a 32-bit opaque value associated with the
      route that can be used for policy decisions such as
      advertisement and filtering of the route.";
  }
}

augment "/rt:routing/rt:ribs/rt:rib"
{
```

```
    description
      "Augment a RIB with statistics.";
    uses rib-statistics;
  }

  augment "/rt:routing/rt:ribs/rt:rib/"
    + "rt:routes/rt:route"
  {
    description
      "Augment a route in RIB with attributes.";
    uses attributes;
  }

  augment "/rt:routing/rt:ribs/rt:rib/"
    + "rt:routes/rt:route/rt:next-hop/rt:next-hop-options/"
    + "rt:simple-next-hop"
  {
    description
      "Augment simple-next-hop with repair-path.";
    uses repair-path;
  }

  augment "/rt:routing/rt:ribs/rt:rib/"
    + "rt:routes/rt:route/rt:next-hop/rt:next-hop-options/"
    + "rt:next-hop-list/rt:next-hop-list/rt:next-hop"
  {
    description
      "Augment the multiple next hops with repair path.";
    uses repair-path;
  }
}
<CODE ENDS>
```

6. 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 pre-configured subset of all available NETCONF or RESTCONF protocol operations and content.

There are a number of data nodes defined in `ietf-rib-extensions.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:

```
/v4ur:next-hop-options/v4ur:simple-next-hop/rib-ext:preference
```

```
/v4ur:next-hop-options/v4ur:simple-next-hop/rib-ext:tag
```

```
/v4ur:next-hop-options/v4ur:next-hop-list/v4ur:next-hop-list  
/v4ur:next-hop/rib-ext:preference
```

```
/v4ur:next-hop-options/v4ur:next-hop-list/v4ur:next-hop-list  
/v4ur:next-hop/rib-ext:tag
```

```
/v6ur:next-hop-options/v6ur:simple-next-hop/rib-ext:preference
```

```
/v6ur:next-hop-options/v6ur:simple-next-hop/rib-ext:tag
```

```
/v6ur:next-hop-options/v6ur:next-hop-list/v6ur:next-hop-list  
/v6ur:next-hop/rib-ext:preference
```

```
/v6ur:next-hop-options/v6ur:next-hop-list/v6ur:next-hop-list  
/v6ur:next-hop/rib-ext:tag
```

For these augmentations to `ietf-routing.yang`, the ability to delete, add, and modify IPv4 and IPv6 static route preference and tag would allow traffic to be misrouted.

Some of the readable data nodes in the `ietf-rib-extensions.yang` module may be considered sensitive or vulnerable in some network environments. It is thus important to control read access (e.g., via `get`, `get-config`, or `notification`) to these data nodes. These are the subtrees and data nodes and their sensitivity/vulnerability:

```
/rt:routing/rt:ribs/rt:rib/rib-ext:statistics
```

```
/rt:routing/rt:ribs/rt:rib/rt:routes/rt:route/rib-ext:metric
```

```
/rt:routing/rt:ribs/rt:rib/rt:routes/rt:route/rib-ext:tag
```

```
/rt:routing/rt:ribs/rt:rib/rt:routes/rt:route /rib-  
ext:application-tag
```

```
/rt:route/rt:next-hop/rt:next-hop-options/rt:simple-next-hop /rib-  
ext:repair-path
```

```
/rt:routes/rt:route/rt:next-hop/rt:next-hop-options /rt:next-hop-  
list/rt:next-hop-list/rt:next-hop/rib-ext:repair-path
```

The exposure of the Routing Information Base (RIB) will expose the routing topology of the network. This may be undesirable since both due to the fact that exposure may facilitate other attacks. Additionally, network operators may consider their topologies to be sensitive confidential data.

All the security considerations for [RFC8349] writable and readable data nodes apply to the augmentations described herein.

7. 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-rib-extension  
Registrant Contact: The IESG.  
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-rib-extension  
namespace: urn:ietf:params:xml:ns:yang:ietf-rib-extension  
prefix: rib-ext  
reference: RFC XXXX
```

8. References

8.1. 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>>.
- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <<https://www.rfc-editor.org/info/rfc3688>>.

- [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>>.
- [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>>.
- [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>>.
- [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>>.
- [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>>.
- [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>>.
- [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>>.
- [RFC8349] Lhotka, L., Lindem, A., and Y. Qu, "A YANG Data Model for Routing Management (NMDA Version)", RFC 8349, DOI 10.17487/RFC8349, March 2018, <<https://www.rfc-editor.org/info/rfc8349>>.

- [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>>.

8.2. Informative References

- [RFC5714] Shand, M. and S. Bryant, "IP Fast Reroute Framework", RFC 5714, DOI 10.17487/RFC5714, January 2010, <<https://www.rfc-editor.org/info/rfc5714>>.
- [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>>.

Appendix A. Combined Tree Diagram

This appendix includes the combined ietf-routing.yang, ietf-ipv4-unicast-routing.yang, ietf-ipv6-unicast-routing.yang and ietf-rib-extensions.yang tree diagram.

```

module: ietf-routing
+--rw routing
  +--rw router-id?                yang:dotted-quad {router-id}?
  +--ro interfaces
    | +--ro interface*           if:interface-ref
  +--rw control-plane-protocols
    | +--rw control-plane-protocol* [type name]
    |   +--rw type                identityref
    |   +--rw name                string
    |   +--rw description?        string
    |   +--rw static-routes
    |     +--rw v4ur:ipv4
    |       +--rw v4ur:route* [destination-prefix]
    |         +--rw v4ur:destination-prefix  inet:ipv4-prefix
    |         +--rw v4ur:description?        string
    |         +--rw v4ur:next-hop
    |           +--rw (v4ur:next-hop-options)
    |             +--:(v4ur:simple-next-hop)
    |               | +--rw v4ur:outgoing-interface?
    |               | | if:interface-ref
    |               | +--rw v4ur:next-hop-address?
    |               | | inet:ipv4-address
    |               | +--rw rib-ext:preference?      uint32
    |               | +--rw rib-ext:tag?             uint32
    |               +--:(v4ur:special-next-hop)
    |               | +--rw v4ur:special-next-hop?   enumeration
    |               +--:(v4ur:next-hop-list)
    |               | +--rw v4ur:next-hop-list

```

```

        +---rw v4ur:next-hop* [index]
            +---rw v4ur:index string
            +---rw v4ur:outgoing-interface?
                | if:interface-ref
            +---rw v4ur:next-hop-address?
                | inet:ipv4-address
            +---rw rib-ext:preference? uint32
            +---rw rib-ext:tag? uint32
+---rw v6ur:ipv6
    +---rw v6ur:route* [destination-prefix]
        +---rw v6ur:destination-prefix inet:ipv6-prefix
        +---rw v6ur:description? string
        +---rw v6ur:next-hop
            +---rw (v6ur:next-hop-options)
                +---:(v6ur:simple-next-hop)
                    +---rw v6ur:outgoing-interface?
                        | if:interface-ref
                    +---rw v6ur:next-hop-address?
                        | inet:ipv6-address
                    +---rw rib-ext:preference? uint32
                    +---rw rib-ext:tag? uint32
                +---:(v6ur:special-next-hop)
                    | +---rw v6ur:special-next-hop? enumeration
                +---:(v6ur:next-hop-list)
                    +---rw v6ur:next-hop-list
                        +---rw v6ur:next-hop* [index]
                            +---rw v6ur:index string
                            +---rw v6ur:outgoing-interface?
                                | if:interface-ref
                            +---rw v6ur:next-hop-address?
                                | inet:ipv6-address
                            +---rw rib-ext:preference? uint32
                            +---rw rib-ext:tag? uint32
+---rw ribs
    +---rw rib* [name]
        +---rw name string
        +---rw address-family identityref
        +---ro default-rib? boolean {multiple-ribs}?
    +---ro routes
        +---ro route* []
            +---ro route-preference? route-preference
            +---ro next-hop
                +---ro (next-hop-options)
                    +---:(simple-next-hop)
                        +---ro outgoing-interface? if:interface-ref
                        +---ro v4ur:next-hop-address? inet:ipv4-address
                        +---ro v6ur:next-hop-address? inet:ipv6-address
                    +---ro rib-ext:repair-path

```



```

        +---ro rib-ext:outgoing-interface?
        |   if:interface-state-ref
        +---ro rib-ext:next-hop-address?
        |   inet:ip-address
        +---ro rib-ext:metric?                               uint32
+---:(special-next-hop)
|   +---ro special-next-hop?                               enumeration
+---:(next-hop-list)
    +---ro next-hop-list
        +---ro next-hop* []
            +---ro outgoing-interface?
            |   if:interface-ref
            +---ro v4ur:address?
            |   inet:ipv4-address
            +---ro v6ur:address?
            |   inet:ipv6-address
            +---ro rib-ext:repair-path
                +---ro rib-ext:outgoing-interface?
                |   if:interface-state-ref
                +---ro rib-ext:next-hop-address?
                |   inet:ip-address
                +---ro rib-ext:metric?                       uint32
+---ro source-protocol                                     identityref
+---ro active?                                             empty
+---ro last-updated?                                       yang:date-and-time
+---ro v4ur:destination-prefix?                           inet:ipv4-prefix
+---ro v6ur:destination-prefix?                           inet:ipv6-prefix
+---ro rib-ext:metric?                                     uint32
+---ro rib-ext:tag*                                        uint32
+---ro rib-ext:application-tag?                           uint32
+---x active-route
    +---w input
        |   +---w v4ur:destination-address?               inet:ipv4-address
        |   +---w v6ur:destination-address?               inet:ipv6-address
    +---ro output
        +---ro route
            +---ro next-hop
                +---ro (next-hop-options)
                    +---:(simple-next-hop)
                        +---ro outgoing-interface?         if:interface-ref
                        +---ro v4ur:next-hop-address?       inet:ipv4-address
                        +---ro v6ur:next-hop-address?       inet:ipv6-address
                    +---:(special-next-hop)
                        |   +---ro special-next-hop?         enumeration
                    +---:(next-hop-list)
                        +---ro next-hop-list
                            +---ro next-hop* []
                                +---ro outgoing-interface?

```

```

|
|
|         if:interface-ref
|         +--ro v4ur:next-hop-address?
|         |         inet:ipv4-address
|         +--ro v6ur:next-hop-address?
|         |         inet:ipv6-address
|         +--ro source-protocol          identityref
|         +--ro active?                  empty
|         +--ro last-updated?            yang:date-and-time
|         +--ro v4ur:destination-prefix? inet:ipv4-prefix
|         +--ro v6ur:destination-prefix? inet:ipv6-prefix
+--rw description?                      string
+--ro rib-ext:statistics
  +--ro rib-ext:total-routes?            uint32
  +--ro rib-ext:total-active-routes?     uint32
  +--ro rib-ext:total-route-memory?      uint64
  +--ro rib-ext:protocol-statistics* []
    +--ro rib-ext:protocol?              identityref
    +--ro rib-ext:routes?                uint32
    +--ro rib-ext:active-routes?         uint32
    +--ro rib-ext:route-memory?          uint64

```

Appendix B. ietf-rib-extension.yang example

The following is an XML example using the RIB extension module and RFC 8349.

```

<routing xmlns="urn:ietf:params:xml:ns:yang:ietf-routing">
  <control-plane-protocols>
    <control-plane-protocol>
      <type>static</type>
      <name>static-routing-protocol</name>
      <static-routes>
        <ipv4 xmlns="urn:ietf:params:xml:ns:yang:\
          ietf-ipv4-unicast-routing">
          <route>
            <destination-prefix>0.0.0.0/0</destination-prefix>
            <next-hop>
              <next-hop-address>192.0.2.2</next-hop-address>
              <preference xmlns="urn:ietf:params:xml:ns:yang:\
                ietf-rib-extension">30</preference>
              <tag xmlns="urn:ietf:params:xml:ns:yang:\
                ietf-rib-extension">99</tag>
            </next-hop>
          </route>
        </ipv4>
        <ipv6 xmlns="urn:ietf:params:xml:ns:yang:\
          ietf-ipv6-unicast-routing">
          <route>

```

```
<destination-prefix>::/0</destination-prefix>
<next-hop>
  <next-hop-address>2001:db8:aaaa::1111</next-hop-address>
  <preference xmlns="urn:ietf:params:xml:ns:yang:\
    ietf-rib-extension">30</preference>
  <tag xmlns="urn:ietf:params:xml:ns:yang:\
    ietf-rib-extension">66</tag>
</next-hop>
</route>
</ipv6>
</static-routes>
</control-plane-protocol>
</control-plane-protocols>
<ribs>
  <rib>
    <name>ipv4-master</name>
    <address-family xmlns:v4ur="urn:ietf:params:xml:ns:yang:\
      ietf-ipv4-unicast-routing">v4ur:ipv4-unicast</address-family>
    <default-rib>true</default-rib>
    <routes>
      <route>
        <destination-prefix xmlns="urn:ietf:params:xml:ns:yang:\
          ietf-ipv4-unicast-routing">0.0.0.0/0</destination-prefix>
        <next-hop>
          <next-hop-address xmlns="urn:ietf:params:xml:ns:yang:\
            ietf-ipv4-unicast-routing">192.0.2.2</next-hop-address>
        </next-hop>
        <route-preference>5</route-preference>
        <source-protocol>static</source-protocol>
        <last-updated>2015-10-24T18:02:45+02:00</last-updated>
      </route>
      <route>
        <destination-prefix xmlns="urn:ietf:params:xml:ns:yang:\
          ietf-ipv4-unicast-routing">198.51.100.0/24\
        </destination-prefix>
        <next-hop>
          <next-hop-address xmlns="urn:ietf:params:xml:ns:yang:\
            ietf-ipv4-unicast-routing">192.0.2.2</next-hop-address>
          <repair-path xmlns="urn:ietf:params:xml:ns:yang:\
            ietf-rib-extension">
            <next-hop-address>203.0.113.1</next-hop-address>
            <metric>200</metric>
          </repair-path>
        </next-hop>
        <route-preference>110</route-preference>
        <source-protocol xmlns:ospf="urn:ietf:params:xml:ns:yang:\
          ietf-ospf">ospf:ospf</source-protocol>
        <last-updated>2015-10-24T18:02:45+02:00</last-updated>
```

```
    </route>
  </routes>
</rib>
<rib>
  <name>ipv6-master</name>
  <address-family xmlns:v6ur="urn:ietf:params:xml:ns:yang:\
    ietf-ipv6-unicast-routing">v6ur:ipv6-unicast</address-family>
  <default-rib>true</default-rib>
  <routes>
    <route>
      <destination-prefix xmlns="urn:ietf:params:xml:ns:yang:\
        ietf-ipv6-unicast-routing">0::/0</destination-prefix>
      <next-hop>
        <next-hop-address xmlns="urn:ietf:params:xml:ns:yang:\
          ietf-ipv6-unicast-routing">2001:db8:aaaa::1111\
        </next-hop-address>
      </next-hop>
      <route-preference>5</route-preference>
      <source-protocol>static</source-protocol>
      <last-updated>2015-10-24T18:02:45+02:00</last-updated>
    </route>
    <route>
      <destination-prefix xmlns="urn:ietf:params:xml:ns:yang:\
        ietf-ipv6-unicast-routing">2001:db8:bbbb::/64\
      </destination-prefix>
      <next-hop>
        <next-hop-address xmlns="urn:ietf:params:xml:ns:yang:\
          ietf-ipv6-unicast-routing">2001:db8:aaaa::1111\
        </next-hop-address>
        <repair-path xmlns="urn:ietf:params:xml:ns:yang:\
          ietf-rib-extension">
          <next-hop-address>2001:db8:cccc::2222</next-hop-address>
          <metric>200</metric>
        </repair-path>
      </next-hop>
      <route-preference>110</route-preference>
      <source-protocol xmlns:ospf="urn:ietf:params:xml:ns:yang:\
        ietf-ospf">ospf:ospf</source-protocol>
      <last-updated>2015-10-24T18:02:45+02:00</last-updated>
    </route>
  </routes>
</rib>
</ribs>
</routing>
```

The following is the same example using JSON format.

```
{
  "ietf-routing:routing": {
    "control-plane-protocols": {
      "control-plane-protocol": [
        {
          "type": "static",
          "name": "static-routing-protocol",
          "static-routes": {
            "ietf-ipv4-unicast-routing:ipv4": {
              "route": [
                {
                  "destination-prefix": "0.0.0.0/0",
                  "next-hop": {
                    "next-hop-address": "192.0.2.2",
                    "ietf-rib-extension:preference": 30,
                    "ietf-rib-extension:tag": 99
                  }
                }
              ]
            },
            "ietf-ipv6-unicast-routing:ipv6": {
              "route": [
                {
                  "destination-prefix": "::/0",
                  "next-hop": {
                    "next-hop-address": "2001:db8:aaaa::1111",
                    "ietf-rib-extension:preference": 30,
                    "ietf-rib-extension:tag": 66
                  }
                }
              ]
            }
          }
        }
      ]
    },
    "ribs": {
      "rib": [
        {
          "name": "ipv4-master",
          "address-family": "ietf-ipv4-unicast-routing:ipv4-unicast",
          "default-rib": true,
          "routes": {
            "route": [
              {
                "next-hop": {
                  "ietf-ipv4-unicast-routing:next-hop-address": \
                    "192.0.2.2"
                }
              }
            ]
          }
        }
      ]
    }
  }
}
```

```
    },
    "route-preference": 5,
    "source-protocol": "static",
    "last-updated": "2015-10-24T18:02:45+02:00",
    "ietf-ipv4-unicast-routing:destination-prefix": \
    "0.0.0.0/0"
  },
  {
    "next-hop": {
      "ietf-rib-extension:repair-path": {
        "next-hop-address": "203.0.113.1",
        "metric": 200
      },
      "ietf-ipv4-unicast-routing:next-hop-address": \
      "192.0.2.2"
    },
    "route-preference": 110,
    "source-protocol": "ietf-ospf:ospf",
    "last-updated": "2015-10-24T18:02:45+02:00",
    "ietf-ipv4-unicast-routing:destination-prefix": \
    "198.51.100.0/24"
  }
]
}
},
{
  "name": "ipv6-master",
  "address-family": "ietf-ipv6-unicast-routing:ipv6-unicast",
  "default-rib": true,
  "routes": {
    "route": [
      {
        "next-hop": {
          "ietf-ipv6-unicast-routing:next-hop-address": \
          "2001:db8:aaaa::1111"
        },
        "route-preference": 5,
        "source-protocol": "static",
        "last-updated": "2015-10-24T18:02:45+02:00",
        "ietf-ipv6-unicast-routing:destination-prefix": "::/0"
      },
      {
        "next-hop": {
          "ietf-rib-extension:repair-path": {
            "next-hop-address": "2001:db8:cccc::2222",
            "metric": 200
          },
          "ietf-ipv6-unicast-routing:next-hop-address": \
```

```
        "2001:db8:aaaa::1111"
      },
      "route-preference": 110,
      "source-protocol": "ietf-ospf:ospf",
      "last-updated": "2015-10-24T18:02:45+02:00",
      "ietf-ipv6-unicast-routing:destination-prefix": \
        "2001:db8:bbbb::/64"
    }
  ]
}
}
```

Appendix C. Acknowledgments

The RFC text was produced using Marshall Rose's `xml2rfc` tool.

The authors wish to thank Les Ginsberg, Krishna Deevi, and Suyoung Yoon for their helpful comments and suggestions.

The authors wish to thank Tom Petch, Rob Wilton and Chris Hopps for their reviews and comments.

Authors' Addresses

Acee Lindem
Cisco Systems
301 Midenhall Way
Cary, NC 27513
United States of America
Email: acee@cisco.com

Yingzhen Qu
Futurewei
2330 Central Expressway
Santa Clara, CA 95050
United States of America
Email: yingzhen.qu@futurewei.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: March 13, 2020

Z. Li
S. Peng
Huawei Technologies
K. LEE
LG U+
September 10, 2019

IPv6 Encapsulation for SFC and IFIT
draft-li-6man-ipv6-sfc-ifit-02

Abstract

Service Function Chaining (SFC) and In-situ Flow Information Telemetry (IFIT) are important path services along with the packets. In order to support these services, several encapsulations have been defined. The document analyzes the problems of these encapsulations in the IPv6 scenario and proposes the possible optimized encapsulation for IPv6.

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 March 13, 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
2. Terminology	3
3. Problem Statement	3
4. Design Consideration	4
4.1. Service Options	4
4.2. IPv6 Service Metadata Options	7
4.2.1. SFC Service Metadata Option	7
4.2.2. IOAM Service Metadata Option	8
4.2.3. IFA Service Metadata Option	8
5. IANA Considerations	9
6. Security Considerations	9
7. References	9
7.1. Normative References	9
7.2. Informative References	11
Authors' Addresses	11

1. Introduction

Service Function Chaining (SFC) [RFC7665] and In-situ Flow Information Telemetry (IFIT) [I-D.song-opsawg-ifit-framework] are important path services along with the packets. In order to support these services, several encapsulations have been defined. Network Service Header (NSH) is defined in [RFC8300] as the encapsulation for SFC. For IFIT encapsulations, In-situ OAM (iOAM) Header is defined in [I-D.ietf-ippm-ioam-data] and Postcard-Based Telemetry (PBT) Header is defined in [I-D.song-ippm-postcard-based-telemetry]. Inband Flow Analyzer (IFA) is also defined in [I-D.kumar-ippm-ifa] to record flow specific information from an end station and/or switches across a network. In the application scenario of IPv6, these encapsulations propose challenges for the data plane. The document analyzes the problems and proposes the possible optimized encapsulation for IPv6.

2. Terminology

SFC: Service Function Chaining

IFIT: In-situ Flow Information Telemetry

IOAM: In-situ OAM

PBT: Postcard-Based Telemetry

IFA: Inband Flow Analyzer

SRH: Segment Routing Header

3. Problem Statement

The problems posed by the current encapsulations for SFC and IFIT in the application scenarios of IPv6 and SRv6 include:

1. According to the encapsulation order recommended in [RFC8200], if the IOAM is encapsulated in the IPv6 Hop-by-Hop options header, in the incremental trace mode of IOAM as the number of nodes traversed by the IPv6 packets increases, the recorded IOAM information will increase accordingly. This will increase the length of the Hop-by-Hop options header and cause increasing difficulties in reading the subsequent Segment Routing Extension Header (SRH) [I-D.ietf-6man-segment-routing-header] and thereby reduce the forwarding performance of the data plane greatly.

2. With the introduction of SRv6 network programming [I-D.ietf-spring-srv6-network-programming], the path services along with the IPv6 packets can be processed at all the IPv6 network nodes or only at the SRv6 enabled network nodes along the path. It is necessary to distinguish the encapsulations for the specific path service which should be processed by the IPv6 path or the SRv6 path.

3. Both NSH and IOAM need the Metadata field to record metadata information. However currently these metadata has to be recorded separately which may generate redundant metadata information or increase the cost of process.

4. There is unnecessary inconsistency in the current encapsulations for IOAM, IFA and PBT in the IPv6 scenario. Especially it seems unnecessary to define a new specific IPv6 header for IFA, i.e. IFA header.

4. Design Consideration

To solve the problems stated above, in the application scenarios of IPv6 and SRv6, the encapsulations of SFC and IFIT can be optimized with the following design considerations:

- o To separate the SFC/IFIT path service into two parts, i.e. instruction and recording parts. The instruction part (normally with fixed length) can be placed in the front IPv6 extension headers including Hop-by-Hop options header, Destination options header, Routing header, etc. while the recording part can be placed in the back IPv6 extension headers such as being placed after IPv6 Routing Header. In this way the path service instruction in the IPv6 extension headers can be fixed as much as possible to facilitate hardware process to keep forwarding performance while the SFC/IFIT metadata recording part is placed afterwards which enables to stop recording when too much recording information has to be carried to reach the limitation of hardware process.
- o To define SFC/IFIT path service instructions as IPv6 options uniformly which can be placed either in the Hop-by-hop options which indicates the path service processed by all IPv6 enabled nodes along the path or in the SRH option TLVs which indicates the path service processed only by the SRv6 nodes along the SRv6 path indicated by the Segment List in the SRH.
- o To define a unified IPv6 metadata header which can be used as a container to record the service metadata of SFC, IFIT and other possible path services.

According to the above design optimization consideration, in the application scenarios of IPv6 and SRv6 the encapsulations for SFC and IFIT can be defined as below.

4.1. Service Options

1. NSH Service Option

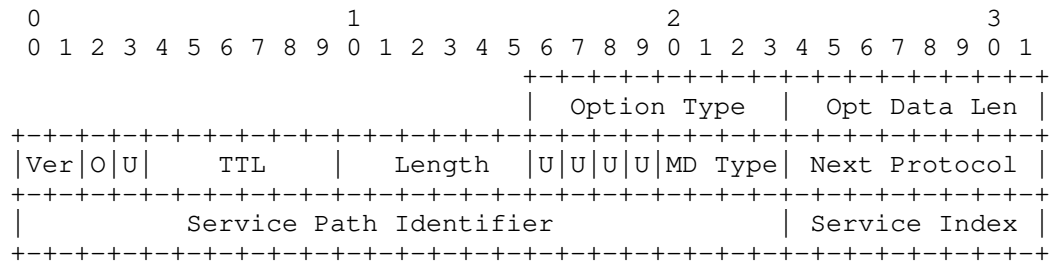


Figure 1. IPv6 Options with NSH instructions

Option Type: TBD_0

Opt Data Len: 8 octets.

Other fields: refer to [RFC8300].

2. IOAM Service Option

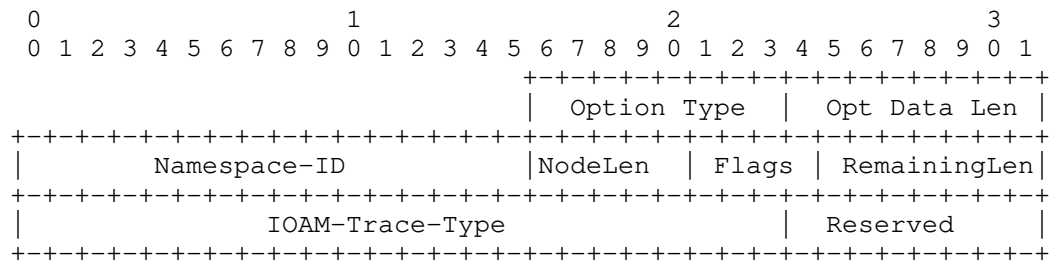


Figure 2. IPv6 Options with IOAM instructions

Option Type: TBD_1

Opt Data Len: 8 octets.

Other fields: refer to [I-D.ietf-ippm-ioam-data].

3. PBT Service Option

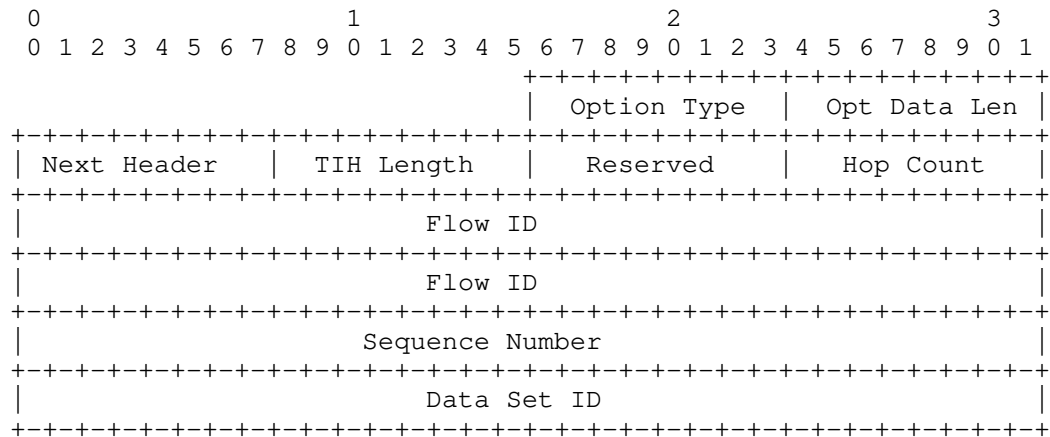


Figure 3. IPv6 Options with PBT instructions

Option Type: TBD_2

Opt Data Len: 20 octets.

Other fields: refer to [I-D.song-ippm-postcard-based-telemetry].

4. IFA Service Option

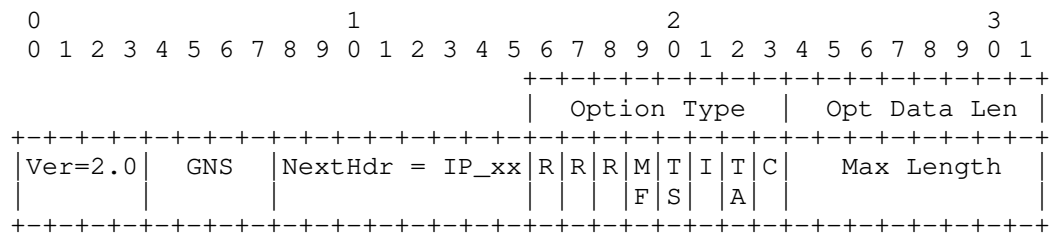


Figure 4. IPv6 Options with IFA instructions

Option Type: TBD_3

Opt Data Len: 4 octets.

Other fields: refer to [I-D.kumar-ippm-ifa].

These options can be put in the IPv6 Hop-by-Hop Options Header or SRH TLV.

4.2. IPv6 Service Metadata Options

As introduced in [I-D.li-6man-enhanced-extension-header], IPv6 Metadata Header is defined as a new type of IPv6 extension header. The metadata is the information recorded by each hop for specific path services, and carried in corresponding service metadata options. The length of the metadata is variable.

4.2.1. SFC Service Metadata Option

For the SFC service, the corresponding SFC service metadata option is defined as shown in Figure 5.

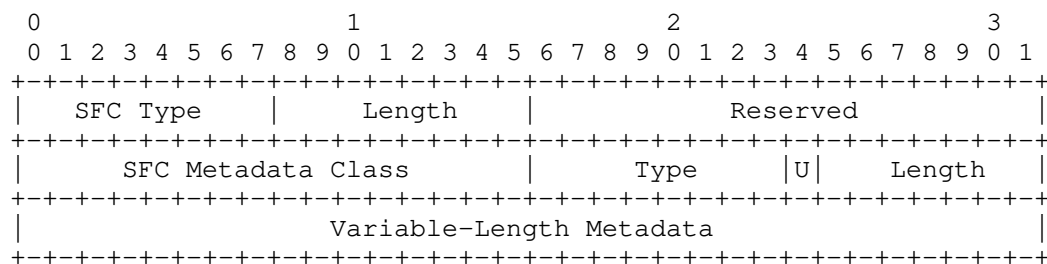


Figure 5. SFC Service Metadata

SFC Type	8-bit identifier of the service type, i.e. SFC. The value is TBD-4.
Length	8-bit unsigned integer. Length of the Service Metadata field, in octets.
Metadata Class	Defines the scope of the Type field to provide a hierarchical namespace. IANA has set up the "NSH MD Class" registry, which contains 16-bit values [RFC8300].
Type	Indicates the explicit type of metadata being carried. The definition of the Type is the responsibility of the MD Class owner.
Unassigned bit	One unassigned bit is available for future use. This bit MUST NOT be set, and it MUST be ignored on receipt.
Length	Indicates the length of the variable-length metadata, in bytes. Detailed specification in [RFC8300].

4.2.2. IOAM Service Metadata Option

For the IOAM service, the corresponding IOAM service metadata option is defined as shown in Figure 6.

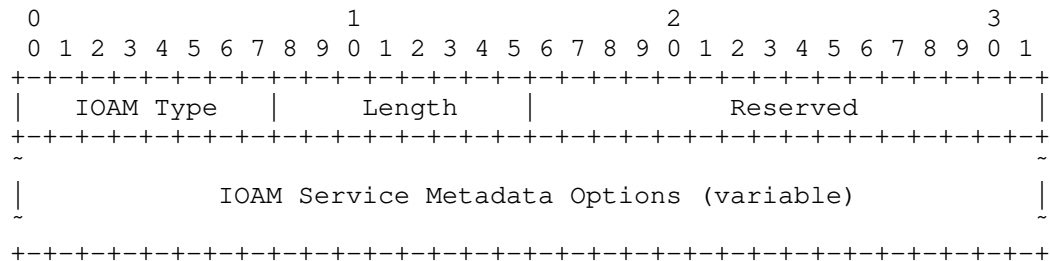


Figure 6. IOAM Service Metadata

IOAM Type	8-bit identifier of the IOAM Service Metadata type. The value is TBD-5.
Length	8-bit unsigned integer. Length of the IOAM Service Metadata field, in octets.
RESERVED	8-bit reserved field MUST be set to zero upon transmission and ignored upon receipt.
IOAM Service Metadata Options	IOAM option data is present as specified by the IOAM Type field, and is defined in Section 4 of [I-D.ietf-ippm-ioam-data].

All the IOAM IPv6 options require 4n alignment. This ensures that 4 octet fields specified in [I-D.ietf-ippm-ioam-data] such as transit delay are aligned at a multiple-of-4 offset from the start of the IPv6 Metadata header.

In addition, to maintain IPv6 extension header 8-octet alignment and avoid the need to add or remove padding at every hop, the Trace-Type for Incremental Tracing Option in IPv6 MUST be selected such that the IOAM node data length is a multiple of 8-octets.

4.2.3. IFA Service Metadata Option

For the IOAM service, the corresponding IOAM service metadata option is defined as shown in Figure 6.

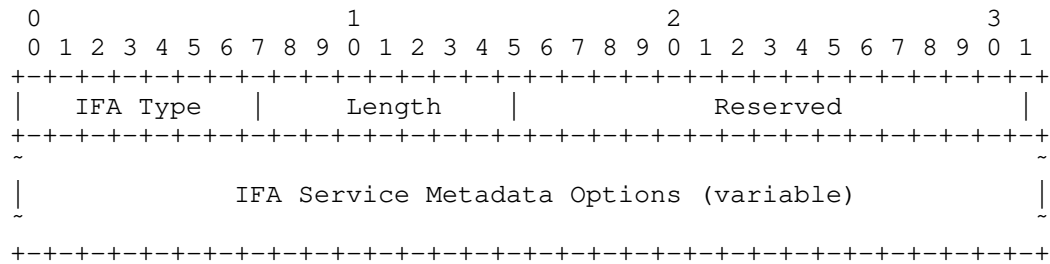


Figure 6. IFA Service Metadata

IFA Type	8-bit identifier of the IFA Service Metadata type. The value is TBD-6.
Length	8-bit unsigned integer. Length of the IOAM Service Metadata field, in octets.
RESERVED	8-bit reserved field MUST be set to zero upon transmission and ignored upon receipt.
IFA Service Metadata Options	IFA option data is present as specified by the IFA Type field.

5. IANA Considerations

Value	Description	Reference
TBD_0	NSH Service Option	[This draft]
TBD_1	IOAM Service Option	[This draft]
TBD_2	PBT Service Option	[This draft]
TBD_3	IFA Service Option	[This draft]
TBD_4	SFC Service Metadata Type	[This draft]
TBD_5	IOAM Service Metadata Type	[This draft]
TBD_6	IFA Service Metadata Type	[This draft]

6. Security Considerations

TBD.

7. References

7.1. Normative References

- [I-D.guichard-spring-nsh-sr]
Guichard, J., Song, H., Tantsura, J., Halpern, J., Henderickx, W., Boucadair, M., and S. Hassan, "NSH and Segment Routing Integration for Service Function Chaining (SFC)", draft-guichard-spring-nsh-sr-01 (work in progress), March 2019.
- [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-22 (work in progress), August 2019.
- [I-D.ietf-ippm-ioam-data]
Brockners, F., Bhandari, S., Pignataro, C., Gredler, H., Leddy, J., Youell, S., Mizrahi, T., Mozes, D., Lapukhov, P., Chang, R., daniel.bernier@bell.ca, d., and J. Lemon, "Data Fields for In-situ OAM", draft-ietf-ippm-ioam-data-06 (work in progress), July 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-01 (work in progress), July 2019.
- [I-D.kumar-ippm-ifa]
Kumar, J., Anubolu, S., Lemon, J., Manur, R., Holbrook, H., Ghanwani, A., Cai, D., Ou, H., and L. Yizhou, "Inband Flow Analyzer", draft-kumar-ippm-ifa-01 (work in progress), February 2019.
- [I-D.song-ippm-postcard-based-telemetry]
Song, H., Zhou, T., Li, Z., Shin, J., and K. Lee, "Postcard-based On-Path Flow Data Telemetry", draft-song-ippm-postcard-based-telemetry-04 (work in progress), June 2019.
- [I-D.song-opsawg-ifit-framework]
Song, H., Li, Z., Zhou, T., Qin, F., Shin, J., and J. Jin, "In-situ Flow Information Telemetry Framework", draft-song-opsawg-ifit-framework-04 (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>>.

- [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>>.
- [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>>.
- [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>>.

7.2. Informative References

- [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>>.

Authors' Addresses

Zhenbin Li
Huawei Technologies
Huawei Bld., No.156 Beiqing Rd.
Beijing 100095
China

Email: lizhenbin@huawei.com

Shuping Peng
Huawei Technologies
Huawei Bld., No.156 Beiqing Rd.
Beijing 100095
China

Email: pengshuping@huawei.com

Kihoon LEE
LG U+
71, Magokjungang 8-ro, Gangseo-gu
Seoul
Republic of Korea

Email: soho8416@lguplus.co.kr

INTERNET-DRAFT
Intended status: Proposed Standard

Z. Li
S. Zhuang
G. Yan
D. Eastlake
Huawei
November 8, 2018

Expires: May 7, 2019

YANG Data Model for Point-to-Point Tunnel Policy
draft-li-rtgwg-tunnel-policy-yang-02

Abstract

This document defines a YANG data model that can be used to configure and manage point-to-point tunnel policy.

Status of this Memo

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

Distribution of this document is unlimited. Comments should be sent to the authors or the TRILL working group mailing list: trill@ietf.org.

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

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

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

Table of Contents

1. Introduction.....	3
2. Definitions and Acronyms.....	3
3. Introduction.....	4
3.1 Tunnel Policy.....	4
3.1.1 Selection Sequence.....	4
3.1.2 Tunnel Binding.....	4
3.2 Tunnel Selector for Routes.....	5
3.3 Tunnel Selector for VPNs.....	6
4. Design of Data Model.....	7
4.1 Tunnel Policy YANG Model.....	7
4.2 Tunnel Selector YANG Model.....	7
5. Tunnel Policy Yang Module.....	9
6. IANA Considerations.....	24
7. Security Considerations.....	24
Acknowledgements.....	25
Informational References.....	26
Normative References.....	26
Authors' Addresses.....	27

1. Introduction

YANG [RFC6020] is a data definition language used to define the contents of a conceptual data store that allows networked devices to be managed using NETCONF [RFC6241]. YANG is proving relevant beyond its initial confines, as bindings to other interfaces (e.g. ReST) and encoding other than XML (e.g. JSON) are being defined. Furthermore, YANG data models can be used as the basis of implementation for other interfaces, such as CLI and programmatic APIs.

This document defines a YANG data model that can be used to configure and manage point-to-point tunnel policy.

2. Definitions and Acronyms

JSON: JavaScript Object Notation

LSP: Label Switched Path

NETCONF: Network Configuration Protocol

RD: Route Distinguisher

TNLM: Tunnel Management

VPN: Virtual Private Network

YANG: A data definition language specified in [RFC6020] for use with NETCONF [RFC6241]

3. Introduction

3.1 Tunnel Policy

Multiple types of tunnels can be used for VPN services, such as LDP LSPs, static LSPs, and CRLSP. It is necessary to select different tunnels for the VPN services to satisfy the required specific tunnel policy.

A tunnel policy determines which type of tunnels can be selected. Tunnel policies can be classified into two modes:

- o Selection Sequence: The system selects a tunnel for the service based on the tunnel type priorities defined in the tunnel policy.
- o Tunnel binding: The system selects only a specified tunnel for the service.

3.1.1 Selection Sequence

Selection sequence, as a tunnel policy mode, specifies the tunnel-selecting sequence and the number of tunnels in the load balancing mode. Selection Sequence is applicable to the tunnels including the LSP, CR-LSP, etc. In selection-sequence mode, tunnels are selected in sequence. If a tunnel listed earlier is Up and not bound, it is selected regardless of whether other services have selected it; if a tunnel is listed later, it is not selected except when load balancing is required or the preceding tunnels are all in the Down state.

3.1.2 Tunnel Binding

Tunnel binding, as a tunnel policy mode, binds a tunnel with a destination IP address. Tunnel binding is only applicable to TE tunnels.

In tunnel binding mode, multiple TE tunnels can be specified to perform load balancing for the same destination IP address. Moreover, the down-switch attribute can be specified to ensure that other tunnels can be selected when all the designated tunnels are unavailable, which keeps the traffic uninterrupted to the maximum extent.

In terms of tunnel selection among TE tunnels, tunnels are selected according to the destination IP address and name of these TE tunnels.

The principles of tunnel selection are as follows:

1. If the tunnel policy designates no TE tunnel for the destination IP address, the tunnels selection sequence is LSP, CR-LSP.
2. If the tunnel policy designates a TE tunnel for the destination IP address, and the designated TE tunnels is available, that TE tunnel is selected.
3. If the tunnel policy designates a TE tunnel for the destination IP address, but the designated TE tunnels is unavailable, the tunnel-selecting result is determined by the down-switch attribute. If the down-switch attribute is configured, another available tunnel is selected based on the sequence of LSP, CR-LSP, and GRE tunnel; if the down-switch attribute is not configured, no tunnel is selected.

3.2 Tunnel Selector for Routes

A tunnel policy selector defines certain matching rules and associates the routes whose attributes matching the rules with specific tunnels. This facilitates flexible tunneling and better satisfies user requirements.

A tunnel policy selector consists of one more policy nodes and the relationship between these policy nodes is "OR". The system checks the policy nodes based on index numbers. If a route matches a policy node in the tunnel policy, the route does not continue to match the next policy node. Each policy node comprises a set of if-match and apply clauses:

1. The if-match clauses define the matching rules that are used to match certain route attributes such as the next hop and RD. The relationship between the if-match clauses of a node is "AND". A route matches a node only when the route meets all the matching rules specified by the if-match clauses of the node.
2. The apply clause specifies actions. When a route matches a node, the apply clause selects a tunnel policy for the route. The matching modes of a node are as follows:
 - a) Permit: If a route matches all the if-match clauses of a node, the route matches the node and the actions defined by the apply clause are performed on the route. If a route does not match one if-match clause of a node, the route continues to match the next node.
 - b) Deny: In this mode, the actions defined by the apply clause

are not performed. If a route matches all the if-match clauses of a node, the route is denied and does not match the next node.

3.3 Tunnel Selector for VPNs

Selection of the tunnel for the VPN services includes the matching rules and the applied tunnel policy. The data model is defined in the drafts of VPN Yang models which are out of the scope of this document. They can refer to the Yang models defined in the document for tunnel policy.

4. Design of Data Model

4.1 Tunnel Policy YANG Model

A tunnel policy determines which type of tunnels can be selected by an application module. The configuration of tunnel policy includes defining the tunnel selection sequence mode and the binding mode for the tunnel selection. The nonexistentCheckFlag controls whether the system allows a nonexistent tunnel policy to be specified in a command.

```

+--rw tnlmGlobal
|   +--rw nonexistentCheckFlag?   boolean
+--rw tunnelPolicys
|   +--rw tunnelPolicy* [tnlPolicyName]
|   |   +--rw tnlPolicyName        string
|   |   +--ro tnlPolicyExist?      tnlPolicyExist
|   |   +--ro tpSubCount?          uint32
|   |   +--rw description?         string
|   |   +--rw tnlPolicyType?       tnlnbaseTnlPolicyType
|   |   +--rw tpNexthops
|   |   |   +--rw tpNexthop* [nexthopIPAddr]
|   |   |   |   +--rw nexthopIPAddr      inet:ipv4-address-no-zone
|   |   |   |   +--rw downSwitch?       boolean
|   |   |   |   +--rw ignoreDestCheck?   boolean
|   |   |   |   +--rw isIncludeLdp?      boolean
|   |   |   |   +--rw tpTunnels
|   |   |   |   |   +--rw tpTunnel* [tunnelName]
|   |   |   |   |   |   +--rw tunnelName    string
|   |   +--rw tnlSelSeqs
|   |   |   +--rw tnlSelSeq!
|   |   |   |   +--rw loadBalanceNum?    uint32
|   |   |   |   +--rw selTnlType1?       tnlnbaseSelTnlType
|   |   |   |   +--rw selTnlType2?       tnlnbaseSelTnlType
|   |   |   |   +--rw selTnlType3?       tnlnbaseSelTnlType
|   |   |   |   +--rw selTnlType4?       tnlnbaseSelTnlType
|   |   |   |   +--rw selTnlType5?       tnlnbaseSelTnlType
|   |   |   |   +--rw selTnlType6?       tnlnbaseSelTnlType
|   |   |   |   +--rw unmix?            boolean

```

4.2 Tunnel Selector YANG Model

A tunnel policy selector defines certain matching rules and associates the routes whose attributes matching the rules with specific tunnels. This facilitates flexible tunneling satisfying user requirements.

Configuration of the tunnel selector and applying it to the BGP VPNv4/VPNv6 address-family can make the VPN service select the specific tunnel for VPN data transmission.

```

+--rw tunnelSelectors
+--rw tunnelSelector* [name]
  +--rw name string
  +--rw tunnelSelectorNodes
    +--rw tunnelSelectorNode* [nodeSequence]
      +--rw nodeSequence uint32
      +--rw matchMode rtpMatchMode
      +--rw matchCondition
        +--rw matchDestPrefixFilters
          +--rw matchDestPrefixFilter!
            +--rw prefixName? string
        +--rw matchIPv4NextHops
          +--rw matchIPv4NextHop!
            +--rw matchType? rtpTnlSelMchType
            +--rw prefixName? string
            +--rw aclNameOrNum? string
        +--rw matchIPv6NextHops
          +--rw matchIPv6NextHop!
            +--rw ipv6PrefixName? string
        +--rw matchCommunityFilters
          +--rw matchCommunityFilter* [cmntyNameOrNum]
            +--rw cmntyNameOrNum string
            +--rw wholeMatch? boolean
            +--rw sortMatch? boolean
        +--rw matchRdFilters
          +--rw matchRdFilter!
            +--rw rdIndex? uint32
      +--rw applyAction
        +--rw applyTnlPolicys
          +--rw applyTnlPolicy!
            +--rw tnlPolicyName? string

augment /bgp:bgp/bgp:global/bgp:afi-safis/bgp:afi-safi/
  bgp:l3vpn-ipv4-unicast:
    +--rw tunnelSelectorName? string
augment /bgp:bgp/bgp:global/bgp:afi-safis/bgp:afi-safi/
  bgp:l3vpn-ipv6-unicast:
    +--rw tunnelSelectorName? string

```

5. Tunnel Policy Yang Module

```

//Tunnel Policy YANG MODEL
<CODE BEGINS> file " tunnel-policy@2018-09-15.yang "
module tunnel-policy {
  namespace "urn:huawei:params:xml:ns:yang:tunnel-policy";
  // replace with IANA namespace when assigned
  prefix tnlp;

  import ietf-bgp {
  prefix bgp;

  }

  import ietf-inet-types {
  prefix inet;
  //rfc6991-Common YANG Data Types
  }

  organization
  "Huawei Technologies Co., Ltd.";
  contact
  "Huawei Industrial Base Bantian, Longgang Shenzhen 518129
   People's Republic of China
   Website: http://www.huawei.com Email: support@huawei.com";
  description
  "This YANG module defines the tunnel policy configuration
  data for tunnel policy service.

  VPN data needs to be carried by tunnels. By default, the
  system selects LSPs to carry VPN services without performing
  load balancing. If this cannot meet the requirements of VPN
  services, a tunnel policy needs to be used. The tunnel policy
  may be a tunnel type prioritizing policy or a tunnel binding
  policy. Determine which type of tunnel policy to use based
  on your actual requirements:
  * A tunnel type prioritizing policy can change the tunnel
  type selected for VPN services and allow load balancing
  among tunnels.
  * A tunnel binding policy can bind a VPN service to
  specified MPLS TE tunnels to provide QoS guarantee for
  the VPN service.

  Terms and Acronyms

  ... ";

  revision 2018-09-15 {
  description
  "Initial revision.";

```

```
}

typedef tnmbaseTnlPolicyType {
type enumeration {
  enum "invalid" {
    description
      "Tunnel policy with null configurations.";
  }
  enum "tnlSelectSeq" {
    description
      "Tunnel select-seq policy. This policy allows you
       to specify the sequence in which different types
       of tunnels are selected and the number of tunnels
       for load balancing.";
  }
  enum "tnlBinding" {
    description
      "Tunnel binding policy. This policy allows you to
       specify the next hop to be bound to a TE tunnel.
       After a TE tunnel is bound to a destination
       address, VPN traffic destined for that destination
       address will be transmitted over the TE tunnel.";
  }
}
description
  "tunnel policy type";
}
typedef tnmbaseSelTnlType {
type enumeration {
  enum "invaild" {
    description
      "Search for invalid tunnels.";
  }
  enum "lsp" {
    description
      "Search for LDP LSPs.";
  }
  enum "cr-lsp" {
    description
      "Search for CR-LSPs.";
  }
  enum "gre" {
    description
      "Search for GREs.";
  }
  enum "ldp" {
    description
      "Search for LDP LSPs.";
  }
  enum "bgp" {
```

```
        description
            "Search for BGP LSPs.";
    }
    enum "srbe-lsp" {
        description
            "Search for SR-LSPs.";
    }
    enum "sr-te" {
        description
            "Search for SR-TE.";
    }
    enum "te" {
        description
            "Search for TE.";
    }
}
description
    "tunnel select type";
}

typedef tnlPolicyExist {
type enumeration {
    enum "true" {
        description
            "The tunnel policy has been configured.";
    }
    enum "false" {
        description
            "The tunnel policy has not been configured.";
    }
}
}
description
    "tunnel policy state";
}

typedef rtpMatchMode {
type enumeration {
    enum "permit" {
        description
            "Matching mode of filters.";
    }
    enum "deny" {
        description
            "Matching mode of filters.";
    }
}
}
description
    "match mode";
}
```

```

typedef rtpTnlSelMchType {
type enumeration {
    enum "matchNHopPF" {
        description
            "Match IPv4 next hops by an IPv4 prefix.";
    }
    enum "matchNHopAcl" {
        description
            "Match IPv4 next hops by an ACL.";
    }
}
description
    "tunnel selector type";
}

```

```

/*

```

A tunnel policy determines which type of tunnels can be selected by an application module.

Tunnel policies can be classified into two modes:

Select-seq: The system selects a tunnel for an application program based on the tunnel type priorities defined in the tunnel policy.

Tunnel binding: The system selects only a specified tunnel for an application program.

The two modes are mutually exclusive.

Configuration example:

```

#
tunnel-policy policy1
    description policy1
    tunnel binding destination 1.1.1.1 te Tunnel0/0/0 down-switch
#
tunnel-policy policy2
    tunnel select-seq cr-lsp gre lsp load-balance-number 2
#
tunnel-policy policy3
    tunnel binding destination 1.1.1.1 te Tunnel0/0/0 down-switch
    tunnel binding destination 3.3.3.3 te Tunnel0/0/0
                                ignore-destination-check
    tunnel binding destination 5.5.5.5 te Tunnel0/0/0
#
*/

```

```

    container tnlmGlobal {
description
    "Global parameters for tunnel policy.";
leaf nonexistentCheckFlag {
    type boolean;
}
}

```

```
    default "true";
    description
        "Nonexistent config check flag of tunnel policy.
        By default, if you specify a nonexistent tunnel policy
        in a command, the command does not take effect. To enable
        the system to allow a nonexistent tunnel policy to be
        specified in a command, run the tunnel-policy
        nonexistent-config-check disable command.";
}
}

container tunnelPolicys {
description
    "List of global tunnel policy configurations. A tunnel
    policy can be used to specify a rule for selecting
    tunnels.";

list tunnelPolicy {
    key "tnlPolicyName";

    description
        "A policy for selecting tunnels to carry services. The
        tunnel management module searches for and returns the
        required tunnels based on the tunnel policy. By default,
        no tunnel policy is configured, the system selects an
        available tunnel in the order of conventional LSPs,
        CR-LSPs, and Local_IFNET LSPs, and load balancing is
        not performed.";

    leaf tnlPolicyName {
        type string {
            length "1..39";
        }
        description
            "Name of a tunnel policy. The value is a string of 1 to
            39 case-sensitive characters, spaces not supported.";
    }
    leaf tnlPolicyExist {
        type tnlPolicyExist;
        config false;
        description
            "Whether a tunnel policy has been configured.";
    }
    leaf tpSubCount {
        type uint32;
        config false;
        description
            "Number of times a tunnel policy is referenced.";
    }
    leaf description {
```



```
    type string {
        length "1..80";
    }
    description
        "Description of a tunnel policy.";
}

leaf tnlPolicyType {
    type tnlnbaseTnlPolicyType;
    default "invalid";
    description
        "Tunnel policy type. The available options are sel-seq,
        binding, and invalid. A tunnel policy can be configured
        with only one policy type.";
}

container tpNexthops {
    must "not(..../tnlPolicyType='tnlBinding') or "
        + "(..../tnlPolicyType='tnlBinding' "
        + "and count(tpNexthop)>=1)";
    description
        "List of tunnel binding configurations.";
    list tpNexthop {
        when "not(..../tnlPolicyType='tnlSelectSeq') or "
            + "(..../tnlPolicyType='tnlBinding'";
        key "nexthopIPAddr";
        max-elements "65535";
        description
            "Rule for binding a TE tunnel to a destination address,
            so that the VPN traffic destined for that destination
            address can be transmitted over the TE tunnel.";
        leaf nexthopIPAddr {
            type inet:ipv4-address-no-zone;
            description
                "Destination IP address to be bound to a tunnel.";
        }
        leaf downSwitch {
            type boolean;
            default "false";
            description
                "Enable tunnel switching. After this option is
                selected, if the bound TE tunnel is unavailable,
                the system will select an available tunnel in
                the order of conventional LSPs, CR-LSPs, and
                Local_IFNET tunnels.";
        }
        leaf ignoreDestCheck {
            type boolean;
            default "false";
            description
                "Do not check whether the destination address of the
```

```

        TE tunnel matches the destination address specified
        in the tunnel policy.";
    }
    leaf isIncludeLdp {
        type boolean;
        must "(../isIncludeLdp='true' and not "
            + "(../downSwitch='true')) or "
            + " ../isIncludeLdp='false'";
        default "false";
        description
            "Is loadbalance with LDP";
    }
    container tpTunnels {
        description
            "List of tunnels available for an application.";
        list tpTunnel {
            key "tunnelName";
            min-elements "1";
            max-elements "16";
            description
                "Tunnel.";
            leaf tunnelName {
                type string {
                    length "1..47";
                }
                description
                    "Name of the specified tunnel.";
            }
        }
    }
}

container tnlSelSeqs {
    when "not(../tnlPolicyType='invalid' or "
        + " ../tnlPolicyType='tnlBinding')";
    must "not(../tnlPolicyType='tnlSelectSeq') or "
        + "(../tnlPolicyType='tnlSelectSeq' and "
        + "count(tnlSelSeq)>=1)";
    description
        "Sequence in which different types of tunnels are
        selected.
        If the value is INVALID, no tunnel type has been
        configured.";
    container tnlSelSeq {
        when "not(../../tnlPolicyType='invalid' or "
            + " ../../tnlPolicyType='tnlBinding') or "
            + " ../../tnlPolicyType='tnlSelectSeq'";
        presence "create tnlSelSeq";
        description
            "Sequence in which different types of tunnels are

```

```
        selected. If the value is INVALID, no tunnel type
        has been configured.";
leaf loadBalanceNum {
    type uint32 {
        range "1..64";
    }
    default "1";
    description
        "Sequence in which different types of tunnels are
        selected. The available tunnel types are CR-LSP,
        and LSP. LSP tunnels refer to LDP LSP tunnels
        here.";
}
leaf selTnlType1 {
    type tnldbbaseSelTnlType;
    default "invalid";
    description
        "Sequence in which different types of tunnels are
        selected. If the value is INVALID, no tunnel type
        has been configured.";
}
leaf selTnlType2 {
    when "not(../selTnlType1='invalid' and "
        + "../tnlPolicyType='tnlSelectSeq' or "
        + "../selTnlType1='invalid')";
    type tnldbbaseSelTnlType;
    default "invalid";
    description
        "Sequence in which different types of tunnels are
        selected. If the value is INVALID, no tunnel type
        has been configured.";
}
leaf selTnlType3 {
    when "not(../selTnlType1='invalid' or "
        + "../selTnlType2='invalid')";
    type tnldbbaseSelTnlType;
    default "invalid";
    description
        "Sequence in which different types of tunnels are
        selected. If the value is INVALID, no tunnel type
        has been configured.";
}
leaf selTnlType4 {
    when "not(../selTnlType1='invalid' or "
        + "../selTnlType2='invalid' or "
        + "../selTnlType3='invalid')";
    type tnldbbaseSelTnlType;
    default "invalid";
    description
        "Sequence in which different types of tunnels are
```

```

        selected. If the value is INVALID, no tunnel type
        has been configured.";
    }
    leaf selTnlType5 {
        when "not (../selTnlType1='invaild' or "
            + "../selTnlType2='invaild' or "
            + "../selTnlType3='invaild' or "
            + "../selTnlType4='invaild')";
        type tnImbaseSelTnlType;
        default "invaild";
        description
            "Sequence in which different types of tunnels are
            selected. If the value is INVALID, no tunnel type
            has been configured.";
    }
    leaf selTnlType6 {
        when "not (../selTnlType1='invaild' or "
            + "../selTnlType2='invaild' or "
            + "../selTnlType3='invaild' or "
            + "../selTnlType4='invaild' or "
            + "../selTnlType5='invaild')";
        type tnImbaseSelTnlType;
        default "invaild";
        description
            "Sequence in which different types of tunnels are
            selected. If the value is INVALID, no tunnel type
            has been configured.";
    }
    leaf unmix {
        type boolean;
        default "false";
        description
            "unmix flag.";
    }
}

}

} //End of container tunnelPolicys

/*
The tunnel selector is specific to BGP/MPLS IP VPN services
(a type of VPN service), selecting a tunnel policy for
VPNv4/VPNv6 routes on the backbone network.

A tunnel selector selects tunnel policies for routes after
filtering routes based on some route attributes such as the
route distinguisher (RD) and next hop. This makes tunnel
selection more flexible.

```

```

A tunnel selector is often used on the autonomous system
boundary router (ASBR) in inter-AS VPN Option B or the
superstratum provider edge (SPE) in hierarchy of VPN (HoVPN).
*/
container tunnelSelectors {
description
  "List of tunnel selectors.";
list tunnelSelector {
  key "name";
  max-elements  "65535";
  description
    "Tunnel selector. Usually used in BGP VPN Option B or
    BGP VPN Option C, tunnel selector selects a proper
    tunnel policy for routes.";

  leaf name {
    type string {
      length "1..40";
    }
    description
      "Name of a tunnel selector. The name is a string of
      1 to 40 case-sensitive characters without spaces.";
  }

  container tunnelSelectorNodes {
    description
      "List of tunnel selector nodes.";
    list tunnelSelectorNode {
      key "nodeSequence";
      min-elements  "1";
      max-elements  "65535";

      leaf nodeSequence {
        type uint32 {
          range "0..65535";
        }
        description
          "Sequence number of a node.
          Specifies the index of a node of the tunnel
          selector.
          When a route-policy is used to filter a route,
          the route first matches the node with the
          smallest node value.";
      }

      leaf matchMode {
        type rtpMatchMode;
        mandatory true;
        description
          "Matching mode of nodes.";
      }
    }
  }
}

```

```
container matchCondition {
  description
    "Match Type List";

  container matchDestPrefixFilters {
    description
      "Match IPv4 destination addresses by the prefix
       filter. The configurations of matching IPv4
       destination addresses by the prefix filter are
       mutually exclusive with the configurations of
       matching IPv4 destination addresses based on
       ACL rules.";

    container matchDestPrefixFilter {
      presence "create matchDestPrefixFilter";
      description
        "Match an IPv4 destination address by the prefix
         filter.";
      leaf prefixName {
        type "string";
        description
          "Name of the specified prefix filter when IPv4
           destination addresses are matched.";
      }
    }
  }
} // End of matchDestPrefixFilters

container matchIPv4NextHops {
  description
    "Match IPv4 next hops by the prefix filter or ACL
     filter. The configurations of matching IPv4 next
     hops by the prefix filter are mutually exclusive
     with the configurations of matching IPv4 next
     hops by the ACL filter.";

  container matchIPv4NextHop {
    presence "create matchIPv4NextHop";
    description
      "Match an IPv4 next hop by the prefix or ACL.";
    leaf matchType {
      type rtpTnlSelMchType;
      description
        "Match type. IPv4 next hops are matched with
         either the prefix or ACL.";
    }
    leaf prefixName {
      when "not (../matchType='matchNHopAcl' or "
        + "not (../matchType)) or "
        + " ../matchType='matchNHopPF' ";
      type "string";
    }
  }
}
```

```

        description
            "Name of the specified prefix when IPv4 next hops
            are matched.";
    }
    leaf aclNameOrNum {
        when "not(..../matchType='matchNHopPF' or "
            + "not(..../matchType)) or "
            + "not(..../matchType='matchNHopAcl'";
        type string {
            length "1..32";
        }
        description
            "Name of the specified ACL when next hops are
            matched, which can be a value ranging from
            2000 to 2999 or a string beginning with a-z
            or A-Z.";
    }
}
} //End of container matchIPv4NextHops

container matchIPv6NextHops {
    description
        "Match IPv6 next hops by the IPv6 prefix filter.";
    container matchIPv6NextHop {
        presence "create matchIPv6NextHop";
        description
            "Match an IPv6 next hop by the IPv6 prefix
            filter.";

        leaf ipv6PrefixName {
            type "string";
            description
                "Name of the specified prefix filter when IPv6
                next hops are matched.";
        }
    }
}
} //End of container matchIPv6NextHops

container matchCommunityFilters {
    description
        "Match community attribute filters.";
    list matchCommunityFilter {
        key "cmntyNameOrNum";
        max-elements "32";
        description
            "Match a community attribute filter.";
        leaf cmntyNameOrNum {
            type string {
                length "1..51";
                pattern '((0*[1-9][0-9]?)|(0*1[0-9][0-9]))|'
            }
        }
    }
}

```

```

        + '([^-9][^?0,50})|'
        + '([[][^?^-9][^?])';
    }
    description
        "Name or index of a community attribute filter.
        It can be a numeral or a string. The ID of a
        basic community attribute filter is an integer
        ranging from 1 to 99; the ID of an advanced
        community attribute filter is an integer
        ranging from 100 to 199. The name of a community
        attribute filter is a string of 1 to 51
        characters. The string cannot contain only
        digits.";
    }
    leaf wholeMatch {
        type boolean;
        default "false";
        description
            "All the communities are matched. It is valid to
            only basic community attribute filters.";
    }
    leaf sortMatch {
        type boolean;
        default "false";
        description
            "Match all community attributes in sequence. It
            is valid to only Advanced community attribute
            filters.";
    }
    }
} //End of container matchCommunityFilters

container matchRdFilters {
    description
        "Match RD filters.";
    container matchRdFilter {
        presence "create matchRdFilter";
        description
            "Match an RD filter.";
        leaf rdIndex {
            type uint32 {
                range "1..1024";
            }
            description
                "Index of an RD filter.";
        }
    }
} //End of container matchRdFilters

} //End of container matchCondition

```



```

        container applyAction {
            description
                "Set Type List";
            container applyTnlPolicys {
                description
                    "Set tunnel policies.";
                container applyTnlPolicy {
                    presence "create applyTnlPolicy";
                    description
                        "Set a tunnel policy.";
                    leaf tnlPolicyName {
                        type string {
                            length "1..39";
                        }
                    }
                    description
                        "Name of a tunnel policy. The name is a
                        string of 1 to 39 case-sensitive characters,
                        spaces not supported.";
                }
            }
        } //End of container applyAction
    }

} //End of container tunnelSelectorNodes

} //End of list tunnelSelector

} //End of container tunnelSelectors

/*
* augment some bgp vpn functions in bgp module.
*/
augment "/bgp:bgp/bgp:global/bgp:afi-safis/" +.....
    "bgp:afi-safi/bgp:l3vpn-ipv4-unicast" {
leaf tunnelSelectorName {
    description
        "Specifies the name of a tunnel selector.";

    type "string";
}
}

augment "/bgp:bgp/bgp:global/bgp:afi-safis/" +.....
    "bgp:afi-safi/bgp:l3vpn-ipv6-unicast" {
leaf tunnelSelectorName {
    description
        "Specifies the name of a tunnel selector.";
}
}

```

```
        type "string";
    }
}
<CODE ENDS>
```

6. IANA Considerations

This document requires no IANA actions.

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:

tbd

Unauthorized access to any data node of these subtrees can adversely affect ... tbd ...

Some of the readable data nodes in this YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control read access (e.g., via get, get-config, or notification) to these data nodes. These are the subtrees and data nodes and their sensitivity/vulnerability:

tbd

Unauthorized access to any data node of these subtrees can disclose ... tbd ...

Acknowledgements

The authors would like to thank the following for their contributions to this work:

Xianping Zhang, Linghai Kong, Xiangfeng Ding, Haibo Wang, and Walker Zheng

Informational References

- [RFC6241] Enns, R., Bjorklund, M., Schoenwaelder, J., and A. Bierman, "Network Configuration Protocol (NETCONF)", RFC 6241, June 2011.
- [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>>.
- [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>>.
- [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>>.
- [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>>.

Normative References

- [RFC6020] Bjorklund, M., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, October 2010.

Authors' Addresses

Zhenbin Li
Huawei Technologies
Huawei Bld., No.156 Beiqing Rd.
Beijing 100095 China

Email: lizhenbin@huawei.com

Shunwan Zhuang
Huawei Technologies
Huawei Bld., No.156 Beiqing Rd.
Beijing 100095 China

Email: zhuangshunwan@huawei.com

Gang Yan
Huawei Technologies
Huawei Bld., No.156 Beiqing Rd.
Beijing 100095 China

Email: yangang@huawei.com

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

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

Copyright and IPR Provisions

Copyright (c) 2018 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. The definitive version of an IETF Document is that published by, or under the auspices of, the IETF. Versions of IETF Documents that are published by third parties, including those that are translated into other languages, should not be considered to be definitive versions of IETF Documents. The definitive version of these Legal Provisions is that published by, or under the auspices of, the IETF. Versions of these Legal Provisions that are published by third parties, including those that are translated into other languages, should not be considered to be definitive versions of these Legal Provisions. For the avoidance of doubt, each Contributor to the IETF Standards Process licenses each Contribution that he or she makes as part of the IETF Standards Process to the IETF Trust pursuant to the provisions of RFC 5378. No language to the contrary, or terms, conditions or rights that differ from or are inconsistent with the rights and licenses granted under RFC 5378, shall have any effect and shall be null and void, whether published or posted by such Contributor, or included with or in such Contribution.

RTGWG Working Group
Internet-Draft
Intended status: Informational
Expires: August 29, 2020

G. Mirsky
ZTE Corp.
February 26, 2020

Identification of Overlay Operations, Administration, and Maintenance
(OAM)
draft-mirsky-rtgwg-oam-identify-04

Abstract

This document analyzes how the presence of Operations, Administration, and Maintenance (OAM) control command and/or special data is identified in some overlay networks and an impact on the choice of identification may have on OAM functionality.

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 August 29, 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	2
2.1. Terminology	2
2.2. Keywords	3
3. A Control Channel in an Overlay Network	3
4. Overlay Network Encapsulations	4
4.1. Encapsulations with Meta-data	4
4.1.1. Available Solutions	6
4.2. Fixed-size Encapsulations	6
4.3. Source Information Availability	7
4.4. On-path OAM	7
5. Conclusions	8
6. IANA Considerations	8
7. Security Considerations	8
8. Acknowledgment	8
9. References	8
9.1. Normative References	9
9.2. Informational References	9
Author's Address	11

1. Introduction

Operations, Administration, and Maintenance (OAM) protocols are used to detect, localize defects in the network, and monitor network performance. Some OAM functions, e.g., failure detection, work in the network proactively, while others, e.g., defect localization, usually performed on-demand. These tasks achieved by a combination of active, passive, and hybrid OAM methods, as defined in [RFC7799].

This document analyzes how the presence of Operations, Administration, and Maintenance (OAM) control command and/or special data, i.e., OAM packet, is identified in some overlay networks, and an impact the choice of identification may have on OAM functionality of active and hybrid OAM methods for the respective overlay network encapsulation.

2. Conventions used in this document

2.1. Terminology

AMM Alternate Marking method

BIER Bit Indexed Explicit Replication

DetNet Deterministic Networks

GUE Generic UDP Encapsulation

HTS Hybrid Two-step

NSH Network Service Header

NVO3 Network Virtualization Overlays

OAM Operations, Administration and Maintenance

SFC Service Function Chaining

TLV Type-Length-Value

VXLAN-GPE Generic Protocol Extension for VXLAN

ACH Associated Channed Header

Underlay Network or Underlay Layer: The network that provides connectivity between the DetNet nodes. MPLS network that provides LSP connectivity between DetNet nodes is an example of an underlay layer.

2.2. Keywords

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

3. A Control Channel in an Overlay Network

There's a need for a general control channel between the endpoints of an overlay network for OAM protocols that can be used for fault detection, diagnostics, maintenance, and other functions. Such a control tunnel is dedicated to carrying only control and management data between tunnel endpoints. In other words, the control channel of an overlay network SHOULD NOT carry the client's data. And the endpoint node SHOULD NOT forward a packet received over the control channel. The identification of the control channel might be using different methods. For example, Virtual Network Identifier might be used to identify the control channel in VXLAN and Geneve.

4. Overlay Network Encapsulations

New overlay network encapsulations analyzed in two groups:

- o encapsulations that support optional meta-data;
- o fixed-size encapsulations.

4.1. Encapsulations with Meta-data

Number of the new encapsulation protocols (e.g., Geneve [I-D.ietf-nvo3-geneve], GUE [I-D.ietf-intarea-gue], and SFC NSH [RFC8300]) support use of Type-Length-Value (TLV) encoding to include optional information into the header. The identification of OAM in these protocols is as the following:

Geneve:

O (1 bit): after the WGLC discussion, the interpretation of the O field has changed. The O field now identifies a control packet. This packet contains a control message. Control messages are sent between tunnel endpoints. Tunnel Endpoints MUST NOT forward the payload and transit devices MUST NOT attempt to interpret it. Since these are infrequent control messages, it is RECOMMENDED that tunnel endpoints direct these packets to a high priority control queue (for example, to direct the packet to a general purpose CPU from a forwarding ASIC or to separate out control traffic on a NIC). Transit devices MUST NOT alter forwarding behavior on the basis of this bit, such as ECMP link selection.

[I-D.mmabb-nvo3-geneve-oam] defines the Geneve encapsulation for active OAM. Initially, four options have been presented:

- + with IP/UDP header demultiplexing active OAM protocols, e.g., Fault Management and Performance Monitoring, can be done using the destination UDP port number.
- + demultiplex active OAM protocols by the value of the Protocol Type field in the Geneve header.
- + with using MPLS Generic Associated Channel Label [RFC5586] and Associated Channel Header (ACH) [RFC4385]. Active OAM protocols are demultiplexed using the value of the Channel Type field.

- + using the new EtherType to identify Geneve OAM and the ACH. Active OAM protocols will be demultiplexed based on the Channel Type field's value.

GUE:

C-bit provides the separate namespace to carry formatted data that are implicitly addressed to the decapsulator to monitor or control the state or behavior of a tunnel. The payload is interpreted as a control message with the type specified in the proto/ctype field. The format and contents of the control message are indicated by the type and can be variable length.

SFC NSH:

0 bit: Setting this bit indicates an OAM packet.

Common between Geneve and NSH is the use of the dedicated flag to identify the OAM packet and, at the same time, the presence of the field that identifies the protocol of the payload that immediately follows after the encapsulation header. [RFC8393] points out that if the value of that field interpreted as none, i.e., no payload follows the header, then OAM may be included in TLVs, thus creating an active OAM packet. The problem with this mechanism to support active OAM methods may be a limitation of the size of data that can be included in a TLV. For example, the maximum size of data in an NSH Meta-data Type 2, as defined in section 2.5.1 [RFC8300], is 512 octets. The maximum length of data in Geneve Option, per section 3.5 [I-D.ietf-nvo3-geneve], is 128 octets. Thus, using one TLV as active OAM packet, would not allow creating test packets of larger size, which is useful when measuring packet loss and latency with synthetic traffic as part of the service activation procedure.

[I-D.ietf-sfc-oam-framework] suggests that the 0 bit used to identify OAM packet and the Next Protocol field identifies the OAM function:

While the presence of OAM marker in the overlay header (e.g., 0 bit in the NSH header) indicates it as OAM packet, it is not sufficient to signal for which OAM function the packet is intended.

At the same time, some of in-situ OAM proposals, e.g., [I-D.ietf-sfc-ioam-nsh], suggest using TLV to communicate hybrid OAM commands and data. The proposed resolution of using the combination of 0 bit and the Next Protocol field:

... the O bit MUST NOT be set for regular customer traffic which also carries IOAM data and the O bit MUST be set for OAM packets which carry only IOAM data without any regular data payload.

implies that the O bit only identifies the active OAM packet and not set when hybrid OAM methods used.

4.1.1. Available Solutions

One of the possible solutions for encapsulations with meta-data has been specified in [I-D.ietf-sfc-multi-layer-oam]:

To identify the active OAM message the value on the Next Protocol field MUST be set to Active SFC OAM. The rules of interpreting the values of O bit and the Next Protocol field are as follows:

- o O bit set and the Next Protocol value is not one of identifying active or hybrid OAM protocol (per [RFC7799] definitions), e.g., defined in this specification Active SFC OAM - a Fixed-Length Context Header or Variable-Length Context Header(s) contain OAM command or data and the type of payload determined by the Next Protocol field;
- o O bit set and the Next Protocol value is one of identifying active or hybrid OAM protocol - the payload that immediately follows SFC NSH contains OAM command or data;
- o O bit is clear - no OAM in a Fixed-Length Context Header or Variable-Length Context Header(s) and the payload determined by the value of the Next Protocol field;
- o O bit is clear, and the Next Protocol value is one of identifying active or hybrid OAM protocol MUST be identified and reported as the erroneous combination. An implementation MAY have control to enable processing of the OAM payload.

From the above-listed rules follows the recommendation to avoid the combination of OAM in a Fixed-Length Context Header or Variable-Length Context Header(s) and in the payload immediately following the SFC NSH because there is no unambiguous way to identify such combination using the O bit and the Next Protocol field.

4.2. Fixed-size Encapsulations

Number of the new encapsulation protocols (e.g., VXLAN-GPE [I-D.ietf-nvo3-vxlan-gpe], BIER [RFC8296]) use fixed-size header. The identification of OAM in these protocols is as the following:

VXLAN-GPE:

OAM Flag Bit (O bit): The O bit is set to indicate that the packet is an OAM packet.

BIER:

OAM packet identified by the value of the Next Protocol field. IANA BIER Next Protocol Identifiers registry includes the identifier for OAM (5).

The use of a combination of OAM Flag Bit and the Next Protocol field in VXLAN-GPE requires clarification of the header interpretation when the OAM Flag Bit is set, and the value of the Next Protocol field is one of defined in section 3.2 of [I-D.ietf-nvo3-vxlan-gpe].

BIER encapsulation, defined in [RFC8296], identifies OAM message immediately following the BIER header by the value of the Next Protocol field.

4.3. Source Information Availability

Availability of the packet originator's source information is required for active two-way OAM, e.g., echo request/reply. In cases when the underlay network is IPv4/IPv6 the source information will be derived from the underlay. But when using MPLS underlay network encapsulation of an active OAM packet have to follow specific rules:

- o if available, use Sender ID in the overlay domain (example BFIR ID in BIER [RFC8296];
- o use IP/UDP encapsulation of an OAM packet in the overlay (similar to Section 4.3 [RFC8029]).

4.4. On-path OAM

In addition to active methods, OAM toolset may include methods that don't use specially constructed and injected in the network test packets. [RFC7799] defines OAM methods that are neither entirely active nor passive but are a combination of both as hybrid methods.

One of the examples of the hybrid OAM methods, in-situ OAM, mentioned in Section 4.1. Another example, Alternate Marking method (AMM) [RFC8321], enables on-path OAM functions, e.g., delay and loss measurements, using the data traffic. Because AMM impact on the network can be minimized, measured metrics can be correlated to the network conditions experienced by the specific service. Of all listed in Section 4, BIER allocated the field that may be used for

AMM, as discussed in [I-D.ietf-bier-pmmm-oam]. Applicability of AMM to other overlay protocols, i.e., SFC NSH discussed in [I-D.mirsky-sfc-pmamm], Geneve [I-D.fmm-nvo3-pm-alt-mark], and in IPv6 networks [I-D.fioccola-v6ops-ipv6-alt-mark], been actively discussed.

Hybrid Two-step (HTS), defined in [I-D.mirsky-ippm-hybrid-two-step], provides on-path collection and transport of the telemetry information. HTS enables accurate and consistent measurements by separating the measurement action from the transporting data while ensuring that the follow-up packet that carries the telemetry information does follow the data packet that had triggered the measurement.

5. Conclusions

OAM control commands and data may be present as part of the overlay encapsulation header or as a payload that follows the overlay network header. The recommendations:

- o OAM in the overlay header, if supported by the overlay network, identified by the dedicated flag. Use of this method as active OAM is possible, but functionality is limited.
- o OAM that follows the overlay header identified as payload type, e.g., by the value of the Next Protocol field.

6. IANA Considerations

This document does not propose any IANA consideration. This section may be removed.

7. Security Considerations

This document lists the OAM requirements for a DetNet domain and does not raise any security concerns or issues in addition to ones common to networking.

8. Acknowledgment

TBD

9. References

9.1. 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>>.
- [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>>.

9.2. Informational References

- [I-D.fioccola-v6ops-ipv6-alt-mark]
Fioccola, G., Velde, G., Cociglio, M., and P. Muley, "IPv6 Performance Measurement with Alternate Marking Method", draft-fioccola-v6ops-ipv6-alt-mark-01 (work in progress), June 2018.
- [I-D.fmm-nvo3-pm-alt-mark]
Fioccola, G., Mirsky, G., and T. Mizrahi, "Performance Measurement (PM) with Alternate Marking in Network Virtualization Overlays (NVO3)", draft-fmm-nvo3-pm-alt-mark-03 (work in progress), October 2018.
- [I-D.ietf-bier-pmmm-oam]
Mirsky, G., Zheng, L., Chen, M., and G. Fioccola, "Performance Measurement (PM) with Marking Method in Bit Index Explicit Replication (BIER) Layer", draft-ietf-bier-pmmm-oam-07 (work in progress), January 2020.
- [I-D.ietf-intarea-gue]
Herbert, T., Yong, L., and O. Zia, "Generic UDP Encapsulation", draft-ietf-intarea-gue-09 (work in progress), October 2019.
- [I-D.ietf-nvo3-geneve]
Gross, J., Ganga, I., and T. Sridhar, "Geneve: Generic Network Virtualization Encapsulation", draft-ietf-nvo3-geneve-14 (work in progress), September 2019.
- [I-D.ietf-nvo3-vxlan-gpe]
Maino, F., Kreeger, L., and U. Elzur, "Generic Protocol Extension for VXLAN", draft-ietf-nvo3-vxlan-gpe-09 (work in progress), December 2019.

- [I-D.ietf-sfc-ioam-nsh]
Brockners, F. and S. Bhandari, "Network Service Header (NSH) Encapsulation for In-situ OAM (IOAM) Data", draft-ietf-sfc-ioam-nsh-02 (work in progress), September 2019.
- [I-D.ietf-sfc-multi-layer-oam]
Mirsky, G., Meng, W., Khasnabish, B., and C. Wang, "Active OAM for Service Function Chains in Networks", draft-ietf-sfc-multi-layer-oam-04 (work in progress), November 2019.
- [I-D.ietf-sfc-oam-framework]
Aldrin, S., Pignataro, C., Kumar, N., Krishnan, R., and A. Ghanwani, "Service Function Chaining (SFC) Operations, Administration and Maintenance (OAM) Framework", draft-ietf-sfc-oam-framework-11 (work in progress), September 2019.
- [I-D.mirsky-ippm-hybrid-two-step]
Mirsky, G., Lingqiang, W., and G. Zhui, "Hybrid Two-Step Performance Measurement Method", draft-mirsky-ippm-hybrid-two-step-04 (work in progress), October 2019.
- [I-D.mirsky-sfc-pmamm]
Mirsky, G., Fioccola, G., and T. Mizrahi, "Performance Measurement (PM) with Alternate Marking Method in Service Function Chaining (SFC) Domain", draft-mirsky-sfc-pmamm-09 (work in progress), December 2019.
- [I-D.mmbb-nvo3-geneve-oam]
Mirsky, G., Xiao, M., Boutros, S., and D. Black, "OAM for use in GENEVE", draft-mmbb-nvo3-geneve-oam-01 (work in progress), January 2020.
- [RFC4385] Bryant, S., Swallow, G., Martini, L., and D. McPherson, "Pseudowire Emulation Edge-to-Edge (PWE3) Control Word for Use over an MPLS PSN", RFC 4385, DOI 10.17487/RFC4385, February 2006, <<https://www.rfc-editor.org/info/rfc4385>>.
- [RFC5586] Bocci, M., Ed., Vigoureux, M., Ed., and S. Bryant, Ed., "MPLS Generic Associated Channel", RFC 5586, DOI 10.17487/RFC5586, June 2009, <<https://www.rfc-editor.org/info/rfc5586>>.
- [RFC7799] Morton, A., "Active and Passive Metrics and Methods (with Hybrid Types In-Between)", RFC 7799, DOI 10.17487/RFC7799, May 2016, <<https://www.rfc-editor.org/info/rfc7799>>.

- [RFC8029] Kompella, K., Swallow, G., Pignataro, C., Ed., Kumar, N., Aldrin, S., and M. Chen, "Detecting Multiprotocol Label Switched (MPLS) Data-Plane Failures", RFC 8029, DOI 10.17487/RFC8029, March 2017, <<https://www.rfc-editor.org/info/rfc8029>>.
- [RFC8296] Wijnands, IJ., Ed., Rosen, E., Ed., Dolganow, A., Tantsura, J., Aldrin, S., and I. Meilik, "Encapsulation for Bit Index Explicit Replication (BIER) in MPLS and Non-MPLS Networks", RFC 8296, DOI 10.17487/RFC8296, January 2018, <<https://www.rfc-editor.org/info/rfc8296>>.
- [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>>.
- [RFC8321] Fioccola, G., Ed., Capello, A., Cociglio, M., Castaldelli, L., Chen, M., Zheng, L., Mirsky, G., and T. Mizrahi, "Alternate-Marking Method for Passive and Hybrid Performance Monitoring", RFC 8321, DOI 10.17487/RFC8321, January 2018, <<https://www.rfc-editor.org/info/rfc8321>>.
- [RFC8393] Farrel, A. and J. Drake, "Operating the Network Service Header (NSH) with Next Protocol "None"", RFC 8393, DOI 10.17487/RFC8393, May 2018, <<https://www.rfc-editor.org/info/rfc8393>>.

Author's Address

Greg Mirsky
ZTE Corp.

Email: gregimirsky@gmail.com

MPLS Working Group
Internet-Draft
Intended status: Informational
Expires: September 8, 2019

L. Andersson
Bronze Dragon Consulting
S. Bryant
A. Malis
Huawei Technologies
N. Leymann
Deutsche Telekom
G. Swallow
Independent
March 7, 2019

Deprecating MD5 for LDP
draft-nslag-mpls-deprecate-md5-04

Abstract

When the MPLS Label Distribution Protocol (LDP) was specified circa 1999, there were very strong requirements that LDP should use a cryptographic hash function to sign LDP protocol messages. MD5 was widely used at that time, and was the obvious choices.

However, even when this decision was being taken there were concerns as to whether MD5 was a strong enough signing option. This discussion was briefly reflected in section 5.1 of RFC 5036 [RFC5036] (and also in RFC 3036 [RFC3036]).

Over time it has been shown that MD5 can be compromised. Thus, there is a concern shared in the security community and the working groups responsible for the development of the LDP protocol that LDP is no longer adequately secured.

This document deprecates MD5 as the signing method for LDP messages. The document also selects a future method to secure LDP messages - the choice is TCP-AO. In addition, we specify that the TBD cryptographic mechanism is to be the default TCP-AO security method.

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 September 8, 2019.

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. Requirement Language	3
2. Background	3
2.1. LDP in RFC 5036	3
2.2. MD5 in BGP	3
2.3. Prior Art	4
3. Securing LDP	4
4. Security Considerations	5
5. IANA Considerations	5
6. Acknowledgements	5
7. References	5
7.1. Normative References	5
7.2. Informative References	6
Authors' Addresses	6

1. Introduction

RFC 3036 was published in January 2001 as a Proposed Standard, and it was replaced by RFC 5035, which is a Draft Standard, in October 2007. Two decades after LDP was originally specified there is a concern shared by the security community and the IETF working groups that develop the LDP protocol that LDP is no longer adequately secured.

LDP currently uses MD5 to cryptographically sign its messages for security security purposes. However, MD5 is a hash function that is no longer considered adequate to meet current security requirements.

1.1. Requirement 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.

2. Background

2.1. LDP in RFC 5036

In Section 5.1 "Spoofing" of RFC 5036 [RFC5036], in list item 2 "Session communication carried by TCP" the following statements are made:

LDP specifies use of the TCP MD5 Signature Option to provide for the authenticity and integrity of session messages.

RFC 2385 [RFC2385] asserts that MD5 authentication is now considered by some to be too weak for this application. It also points out that a similar TCP option with a stronger hashing algorithm (it cites SHA-1 as an example) could be deployed. To our knowledge, no such TCP option has been defined and deployed. However, we note that LDP can use whatever TCP message digest techniques are available, and when one stronger than MD5 is specified and implemented, upgrading LDP to use it would be relatively straightforward.

2.2. MD5 in BGP

There has been a similar discussion among working groups developing the BGP protocol. BGP has already replaced MD5 with TCP-AO. This was specified in RFC 7454 [RFC7454].

To secure LDP the same approach will be followed, TCP-AO will be used for LDP also.

As far as we are able to ascertain, there is currently no recommended, mandatory to implement, cryptographic function specified. We are concerned that without such a mandatory function, implementations will simply fall back to MD5 and nothing will really be changed. The MPLS working group will need the expertise of the

security community to specify a viable security function that is suitable for wide scale deployment on existing network platforms.

2.3. Prior Art

RFC 6952 [RFC6952] discusses a set of routing protocols that all are using TCP for transport of protocol messages, according to guidelines set forth in Section 4.2 of "Keying and Authentication for Routing Protocols Design Guidelines", RFC 6518 [RFC6518].

RFC 6952 takes a much broader approach than this document, it discusses several protocols and also securing the LDP session initialization. This document has a narrower scope, securing LDP session messages only. LDP in initialization mode is addressed in RFC 7349 [RFC7349].

RFC 6952 and this document, basically suggest the same thing, move to TCP-AO and deploy a strong cryptographic algorithm.

All the protocols discussed in RFC 6952 should adopt the approach to securing protocol messages over TCP.

3. Securing LDP

Implementations conforming to this RFC MUST implement TCP-AO to secure the TCP sessions carrying LDP in addition to the currently required TCP MD5 Signature Option.

A TBD cryptographic mechanism must be implemented and provided to TCP-AO to secure LDP messages.

The TBD mechanism is the preferred option, and MD5 SHOULD only be used when TBD is unavailable.

Note: The authors are not experts on this part of the stack, but it seems that TCP security negotiation is still work in progress. If we are wrong, then we need to include a requirement that such negotiation is also required. In the absence of a negotiation protocol, however, we need to leave this as a configuration process until such time as the negotiation protocol work is complete. On completion of a suitable negotiation protocol we need to issue a further update requiring its use.

Cryptographic mechanisms do not have an indefinite lifetime, the IETF hence anticipates updating default cryptographic mechanisms over time.

The TBD default security function will need to be chosen such that it can reasonably be implemented on a typical router route processor, and which will provide adequate security without significantly degrading the convergence time of a Label Switching Router (LSR).

Without a function that does not significantly impact router convergence we simply close one vulnerability and open another.

Note: As experts on the LDP protocol, but not on security mechanisms, we need to ask the security area for a review of our proposed approach, and help correcting any misunderstanding of the security issues or our misunderstanding of the existing security mechanisms. We also need a recommendation on a suitable security function (TBD in the above text).

4. Security Considerations

This document is entirely about LDP operational security. It describes best practices that one should adopt to secure LDP messages and the TCP based LDP sessions between LSRs.

This document does not aim to describe existing LDP implementations, their potential vulnerabilities, or ways they handle errors. It does not detail how protection could be enforced against attack techniques using crafted packets.

5. IANA Considerations

There are no requests for IANA actions in this document.

Note to the RFC Editor - this section can be removed before publication.

6. Acknowledgements

-

-

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>>.

- [RFC2385] Heffernan, A., "Protection of BGP Sessions via the TCP MD5 Signature Option", RFC 2385, DOI 10.17487/RFC2385, August 1998, <<https://www.rfc-editor.org/info/rfc2385>>.
- [RFC5036] Andersson, L., Ed., Minei, I., Ed., and B. Thomas, Ed., "LDP Specification", RFC 5036, DOI 10.17487/RFC5036, October 2007, <<https://www.rfc-editor.org/info/rfc5036>>.
- [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>>.

7.2. Informative References

- [RFC3036] Andersson, L., Doolan, P., Feldman, N., Fredette, A., and B. Thomas, "LDP Specification", RFC 3036, DOI 10.17487/RFC3036, January 2001, <<https://www.rfc-editor.org/info/rfc3036>>.
- [RFC6518] Lebovitz, G. and M. Bhatia, "Keying and Authentication for Routing Protocols (KARP) Design Guidelines", RFC 6518, DOI 10.17487/RFC6518, February 2012, <<https://www.rfc-editor.org/info/rfc6518>>.
- [RFC6952] Jethanandani, M., Patel, K., and L. Zheng, "Analysis of BGP, LDP, PCEP, and MSDP Issues According to the Keying and Authentication for Routing Protocols (KARP) Design Guide", RFC 6952, DOI 10.17487/RFC6952, May 2013, <<https://www.rfc-editor.org/info/rfc6952>>.
- [RFC7349] Zheng, L., Chen, M., and M. Bhatia, "LDP Hello Cryptographic Authentication", RFC 7349, DOI 10.17487/RFC7349, August 2014, <<https://www.rfc-editor.org/info/rfc7349>>.
- [RFC7454] Durand, J., Pepelnjak, I., and G. Doering, "BGP Operations and Security", BCP 194, RFC 7454, DOI 10.17487/RFC7454, February 2015, <<https://www.rfc-editor.org/info/rfc7454>>.

Authors' Addresses

Loa Andersson
Bronze Dragon Consulting

Email: loa@pi.nu

Stewart Bryant
Huawei Technologies

Email: stewart.bryant@gmail.com

Andrew G. Malis
Huawei Technologies

Email: agmalis@gmail.com

Nicolai Leymann
Deutsche Telekom

Email: N.Leymann@telekom.de

George Swallow
Independent

Email: swallow.ietf@gmail.com

Network Working Group
Internet-Draft
Intended status: Informational
Expires: August 2, 2019

F. Templin, Ed.
G. Saccone
Boeing Research & Technology
G. Dawra
LinkedIn
A. Lindem
V. Moreno
Cisco Systems, Inc.
January 29, 2019

Scalable De-Aggregation for Overlays Using the Border Gateway Protocol
(BGP)
draft-templin-rtgwg-scalable-bgp-01.txt

Abstract

The Border Gateway Protocol (BGP) has well-known limitations in terms of the numbers of routes that can be carried and stability of the routing system. This is especially true when mobile nodes frequently change their network attachment points, which in the past has resulted in excessive announcements and withdrawals of de-aggregated prefixes. This document discusses a means of accommodating scalable de-aggregation of IPv6 prefixes for overlay networks using BGP.

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 August 2, 2019.

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
2. Overview and Analysis	2
3. Opportunities and Limitations	4
4. Use Cases	4
5. Implementation Status	4
6. IANA Considerations	5
7. Security Considerations	5
8. Acknowledgements	5
9. References	5
9.1. Normative References	5
9.2. Informative References	5
Appendix A. Change Log	6
Authors' Addresses	6

1. Introduction

The Border Gateway Protocol (BGP) [RFC4271] has well-known limitations in terms of the numbers of routes that can be carried and the stability of the routing system. This is especially true for routing systems that include mobile nodes that frequently change their network attachment points, which in the past have resulted in excessive announcements and withdrawals of de-aggregated prefixes. This document discusses a means of accommodating scalable de-aggregation of IPv6 prefixes [RFC8200] for overlay networks using BGP.

2. Overview and Analysis

As discussed in [I-D.ietf-rtgwg-atn-bgp] and [I-D.templin-intarea-6706bis], the method for accommodating de-aggregation is to institute an overlay network instance of BGP that is separate and independent from the global Internet BGP routing system. The overlay is presented to the global Internet as a small number of aggregated IPv6 prefixes (also known as Mobility Service Prefixes (MSPs)) that never change. In this way, the Internet BGP routing system sees only stable aggregated MSPs (e.g., 2001:db8::/32)

and is completely unaware of any de-aggregation or mobility-related churn that may be occurring within the overlay.

The overlay is operated by an Overlay Service Provider (OSP), and consists of a core Autonomous System (AS) with core AS Border Routers (c-ASBRs) that connect to stub ASes with stub ASBRs (s-ASBRs) in a hub-and-spokes fashion. Mobile nodes associate with nearby (i.e., regional) stub ASes for extended timeframes, and change to new stub ASes only after movements of significant topological or geographical distance. Mobility-related changes between stub ASes are therefore normally infrequent.

The s-ASBRs use eBGP to announce de-aggregated Mobile Network Prefixes (MNPs) of mobile nodes (e.g., 2001:db8:1:2::/64, etc.) to their neighboring c-ASBRs, but do not announce fine-grained mobility events such as a mobile node moving to a new network attachment point. Instead, mobile nodes coordinate with stub ASes using mobility protocols such as MIPv6, LISP, AERO, etc. and stub ASes accommodate these localized mobility events without disturbing the c-ASBRs.

The c-ASBRs originate "default" to their neighboring s-ASBRs but do not announce any MNP routes. In this way, MNP announcements and withdrawals are unidirectional from s-ASBRs to c-ASBRs only, thereby suppressing BGP updates on the reverse path. The c-ASBRs in turn use iBGP to maintain a consistent view of the full topology. BGP Route Reflectors (RRs) [RFC4456] can also be used to support increased c-ASBR scaling.

Each c-ASBR should be able to carry at least as many routes as a typical core router in the global public Internet BGP routing system. Since the number of active routes in the Internet is rapidly approaching 1 million (1M), viable c-ASBRs must be capable of carrying at least 1M MNP routes (this has been proven even for BGP running on lightweight virtual machines). The method for increasing scaling therefore is to divide the MSP into longer sub-MSPs, and to assign a different set of c-ASBRs for each sub-MSP.

For example, the MSP 2001:db8::/32 could be sub-divided into sub-MSPs such as 2001:db8:0010::/44, 2001:db8:0020::/44, 2001:db8:0030::/44, etc. with each sub-MSP assigned to a different set of c-ASBRs. Each s-ASBR peers with at least one member of each c-ASBR set and uses route filters such that BGP updates are only sent to the c-ASBR(s) that aggregate the specific sub-MSP. Then, assuming 1 thousand (1K) or more sub-MSPs (each with its own set of c-ASBRs) the entire BGP overlay routing system should be able to service 1 billion (1B) MNPs or more.

3. Opportunities and Limitations

Since a lightweight virtual machine (e.g., a linux image running quagga in the cloud) can service up to 1M MNPs using BGP, it is likely that dedicated high-performance IPv6 router hardware could support even more. With such dedicated high-performance hardware, the number of MNPs could be increased further.

The deployed numbers of s-ASBRs even for very large overlays should not exceed a c-ASBR's capacity for BGP peering sessions. For example, c-ASBRs should be capable of servicing 1K or more BGP peering sessions, with the upper bound limited by keepalive and update control messaging overhead. Conversely, s-ASBRs should be capable of supporting even more sessions since they only receive keepalives and only send updates for mobile nodes within their local stub ASes.

Mobile nodes should refrain from moving rapidly between stub ASes for no good reason, since the objective is only to reduce routing stretch due to movement of significant distances. OSPs could employ disincentives such as surcharge penalties for gratuitous mobility, but intentional abuse would also yield little reward since only the bad actor (i.e., and not others) would be subject to MNP instability.

Packets sent between mobile nodes that associate with different stub ASes would initially need to be forwarded through the core AS, which presents a forwarding bottleneck. For this reason, a route optimization function is needed to reduce congestion in the core. Since c-ASBRs should be commercial off-the-shelf (COTS) dedicated high-performance IPv6 routers, however, they should not be required to participate directly in any out-of-band route optimization signaling. Instead, route optimization should be coordinated by stub AS network elements and/or the mobile nodes themselves.

4. Use Cases

Use cases include Unmanned Air Systems (UAS) in controlled and uncontrolled airspaces, Intelligent Transportation Systems (ITS) in urban air/ground mobility environments, aviation networks, enterprise mobile device users, and cellular network users. Any other use cases in which an OSP services large numbers of mobile nodes are also in scope.

5. Implementation Status

The arrangement of stub and core ASes described in this document has been implemented using standards-compliant linux operating systems and BGP routing protocol implementations (i.e., quagga). No new code

was included, and all requirements were satisfied through standard configuration options.

6. IANA Considerations

This document does not introduce any IANA considerations.

7. Security Considerations

Security considerations are discussed in the references.

8. Acknowledgements

This work is aligned with the FAA as per the SE2025 contract number DTFAWA-15-D-00030.

This work is aligned with the NASA Safe Autonomous Systems Operation (SASO) program under NASA contract number NNA16BD84C.

This work is aligned with the Boeing Information Technology (BIT) MobileNet program.

9. References

9.1. Normative References

- [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>>.
- [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>>.

9.2. Informative References

- [I-D.ietf-rtgwg-atn-bgp] Templin, F., Saccone, G., Dawra, G., Lindem, A., and V. Moreno, "A Simple BGP-based Mobile Routing System for the Aeronautical Telecommunications Network", draft-ietf-rtgwg-atn-bgp-01 (work in progress), January 2019.

[I-D.templin-intarea-6706bis]

Templin, F., "Asymmetric Extended Route Optimization (AERO)", draft-templin-intarea-6706bis-03 (work in progress), December 2018.

Appendix A. Change Log

<< RFC Editor - remove prior to publication >>

Changes from -00 to -01:

- o added Route Reflectors
- o introduced term "Overlay Service Provider (OSP)"
- o removed estimate of number of routes for high-performance routers
- o revised text on route optimization
- o added use case and implementation sections

Status as of 01/23/2018:

- o -00 draft published

Authors' Addresses

Fred L. Templin (editor)
Boeing Research & Technology
P.O. Box 3707
Seattle, WA 98124
USA

Email: fltemplin@acm.org

Greg Saccone
Boeing Research & Technology
P.O. Box 3707
Seattle, WA 98124
USA

Email: gregory.t.saccone@boeing.com

Gaurav Dawra
LinkedIn
USA

Email: gdawra.ietf@gmail.com

Acee Lindem
Cisco Systems, Inc.
USA

Email: acee@cisco.com

Victor Moreno
Cisco Systems, Inc.
USA

Email: vimoreno@cisco.com

RTWG Working Group
Internet-Draft
Intended status: Standards Track
Expires: December 31, 2019

S. Wood, Ed.
Cisco Systems
B. Wu, Ed.
Q. Wu, Ed.
Huawei
C. Menezes
HPE Aruba
June 29, 2019

YANG Data Model for SD-WAN OSE service delivery
draft-wood-rtgwg-sdwan-ose-yang-01

Abstract

This document defines two SD-WAN OSE Open SD-WAN Exchange (OSE) service YANG modules to enable the orchestrator in the enterprise network to implement SD-WAN inter-domain reachability and connectivity services and application aware traffic steering services.

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 December 31, 2019.

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 diagram	4
2. Terminology	4
3. Definitions	4
4. The SD-WAN OSE Service Model Requirements	5
4.1. Reachability & Route Exchange Requirements	5
4.2. Network Segmentation Requirements	5
4.3. Path Management Requirements	6
5. Service Model Usage	6
6. Design of the Data Model	8
6.1. OSE Gateway Service Model	8
6.2. OSE Path Service Model	10
7. SD-WAN OSE Gateway Service YANG Module	13
8. SD-WAN OSE Path Service YANG Module	17
9. Security Considerations	27
10. IANA Considerations	28
11. References	29
11.1. Normative References	29
11.2. Informative References	29
Appendix A. Acknowledges	30
Authors' Addresses	30

1. Introduction

Software-Defined WAN networking (SDWAN) has become a major new technology in Wide Area Networking. SDWAN architecture is a combination of data and control plane orchestration, proprietary control-plane enhancements as well as single-hop, CE-CE data-planes often referred to as "fabrics". On top of this infrastructure, centralized network policy administration and distribution is provided to achieve a specific set of network outcomes or use-cases.

As a result of the use-case driven approach, SDWAN technology solutions often encode choices about data-plane and protocol operation into associated data-plane, control-plane and controller subsystems. These choices are intended to simplify deployment of SDWAN use-cases, but often result in systems that are not compatible and network elements that cannot interoperate in the manner of traditional, standards-based IP networks.

The Open SD-WAN Exchange (OSE) is an open framework to allow for one vendor SD-WAN domain to communicate with another vendor SD-WAN domain. The goal is to enable interworking between different SDWAN domains via the definition of standard service behaviours as well as standard data models to define those services. The underlying service implementation in each domain is only relevant in that it meets the specified service definition. To create OSE SD-WAN services across domain, a higher layer orchestrator may use generic API calls based on the service models to create the desired SDWAN services within each domain via the serving SDWAN manager.

The services currently defined by specification [OSE] include:

- o OSE Gateway Reachability services
- o Application Path Management Services

This document defines two SD-WAN service YANG modules to enable the orchestrator in the enterprise network to implement SD-WAN inter-domain reachability and connectivity services and application aware traffic steering services. The SD-WAN OSE Service Model is for enterprise own network.

1.1. Terminology

The following terms are defined in [RFC6241] and are not redefined here:

- o client
- o server
- o configuration data
- o state data

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 diagram

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

2. Terminology

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 [RFC2119].

3. Definitions

This document uses the following terms:

Service Provider (SP): The organization (usually a commercial undertaking) responsible for operating the network that offers VPN services to clients and customers.

Customer Edge (CE) Device: Equipment that is dedicated to a particular customer and is directly connected to one or more PE devices via attachment circuits. A CE is usually located at the customer premises, and is usually dedicated to a single VPN, although it may support multiple VPNs if each one has separate attachment circuits. The CE devices can be routers, bridges, switches or hosts.

Provider Edge (PE) Device: Equipment managed by the SP that can support multiple VPNs for different customers, and is directly connected to one or more CE devices via attachment circuits. A PE is usually located at an SP Point of Presence (PoP) and is managed by the SP.

SDWAN Manager: SDWAN Manager is the domain specific manager and controller required to configure, manage and control a particular SDWAN domain. To create OSE SDWAN services, a higher layer orchestrator may use OSE defined API calls to create the desired SDWAN services within each domain via the serving SDWAN manager.

Client Orchestration: The Client Orchestration layer is an abstraction of a service level orchestrator or software that implements control the the SDWAN through the defined OSE APIs. The OSE service specifications do not specify the functions and procedures within this entity apart from the fact that it would use the OSE APIs. The client orchestration layer is a functional block which would implement OSE API calls to one or more serving SDWAN managers.

SD-WAN controller: The SD-WAN Controller is a reference block that encompasses the network control-plane functions required to operate the SDWAN network. The SD-WAN network controller delivers control-plane/data-plane separation the is the realization of SDN architecture within the SD-WAN usecase. Each SD-WAN network controller is managed and configured by the SD-WAN manager. The interface between SDWAN network controller and SD-WAN network manager for this purpose is currently outside the scope of the document.

4. The SD-WAN OSE Service Model Requirements

This section provides a common definition for service types required across different SD-WAN vendor domains. The Open SD-WAN Exchange (OSE) model focuses on interoperability between domains, rather than specifying standard protocol and operations with each SD-WAN domain.

The OSE interoperability models focus on the definition of a standard set of service models and parameters that can be implemented in an SDWAN management system to configure interoperable services within an SDWAN domain and between SDWAN domains.

4.1. Reachability & Route Exchange Requirements

In [OSE]SD-WAN reference model, it is assumed that communication between sites in different domains happening via the OSE gateway which suggests that traffic spanning the domains will be backhauled to the OSE gateway. The interfaces between the gateways are called NNI interfaces. The interconnection between OSE gateways includes the following:

- o OSE gateway interconnection: There may be multiple links between OSE gateways. To mitigate the constraints of the underlying network between the OSE gateway, an IP overlay tunnel needs to be established, and provide simple configuration and operation. It is assumed that GRE and IKE based IPsec can be used.
- o Route exchange: Provides L3 reachability information exchange to facilitate L3 connectivity between SD-WAN domains.

4.2. Network Segmentation Requirements

In addition to the basic connection, the inter-domain interconnection needs to ensure the interworking of network segments. Network segmentation divides an enterprise network into different traffic or routing contexts to provide clear separation of traffic of each segment. These are often referred to as Virtual networks. The most common technology of network segmentation are virtual LANs, or VLANs,

for Layer 2 implementation, and virtual routing and forwarding, or VRF, for Layer 3 implementation. For traffic flowing across SD-WAN domains boundaries, segmentation must be preserved. A method of configuration is required to ensure per segment traffic flow separation while passing through SD-WAN domain boundaries. Such use case is also described in Augmenting RFC4364 Technology to Provide Secure Layer L3VPNs over Public Infrastructure [I-D.rosen-bess-secure-l3vpn]. Therefore, as specified in BGP/MPLS IP VPN [RFC4364] for Multi-AS use cases, it is assumed that MP-BGP with Option B is preferred due to its ease for provisioning, segmentation and operations. For some cases when Option B is not available, separate instances of BGP to be configured on a per VRF basis, which is Option A. This may require more involvement from the provisioning systems.

4.3. Path Management Requirements

As specified in ONUG SD-WAN whitepaper[ONUG], dynamic path selection is one of the core features of the SD-WAN, which site-to-site packets can be distributed across multiple WAN connections in real-time, based on current link metrics such as delay, loss and jitter. In this model, a path is considered to be an access network and not a path within an access network, although the latter is not precluded. For business critical applications traversing SD-WAN domains, policies via standardized APIs need to be provisioned to guarantee end-to-end SLA requirements and each domain is responsible for implementing consistent policy enforcement behaviour. Since inter-domain traffic are all backhauled by the OSE gateways, each part of the traversing path needs to be consistent.

Note: A method needs to be specified for budgeting end-to-end delay across multiple domains - delay/loss/jitter needs to be shared so that each domain can compute the total path, determine who's violating and then execute path change.

5. Service Model Usage

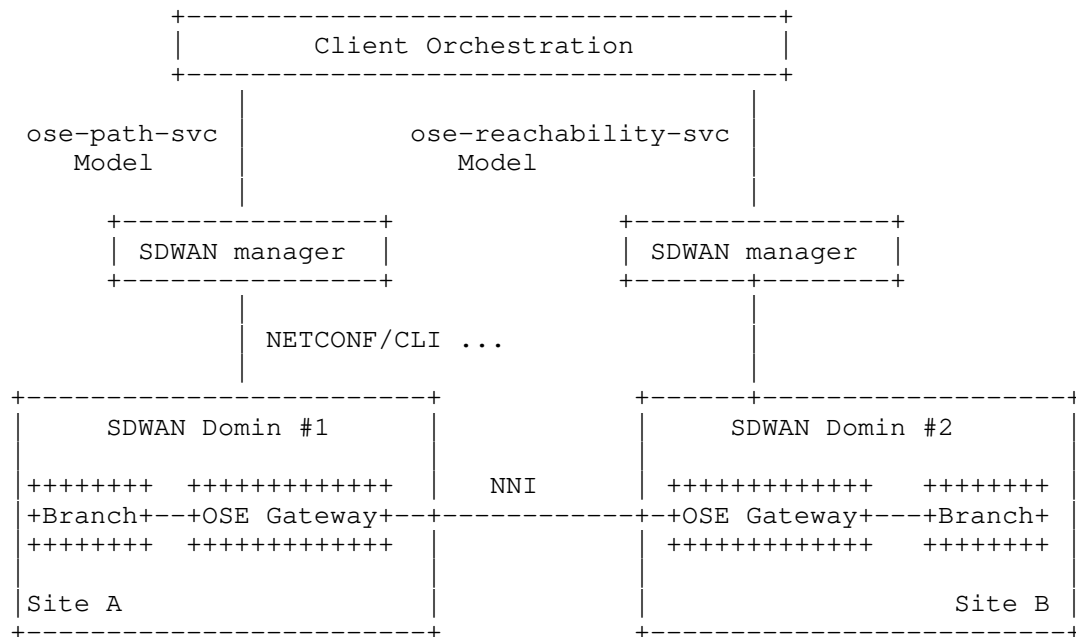


Figure 1

As shown in figure 1, communication between branch sites sitting in domain#1 and domain#2 happens via a border element referred to as the OSE Gateway. This border element interworks the SDWAN control and data plane of the SDWAN domain to a common, defined NNI carrying routing information to establish reachability between domains. It also carries segmentation identifiers that are mutually agreed and configured within each OSE gateway by the domain serving SDWAN manager. The serving SDWAN manager in each respective domain is configured by the operator with information about which segments in each domain are to be connected.

Segment connections must be 1:1 across each OSE gateway.

Note: The detailed control and data plane specifications for the OSE Gateway NNI will refer to the definition of the relevant SD-WAN protocols in the IETF.

The ONUG SD-WAN service YANG model provides an abstracted interface to configure, and manage the components of an SD-WAN service. The components of the SD-WAN service include the OSE Gateway Service component and the Path Management Service component. OSE gateway service component defines Reachability and Route Exchange Segmentation requirements for OSE Gateway devices while path

management service component defines path management policy for domain serving SD-WAN managers.

A typical usage for this model is to generate Restconf[RFC8040] API used between Client Orchestrator layer and SDWAN manager and used by an enterprise operator to provision the inter-domain services. Before configuring the inter-domain path management policy service, the ose-reachability-svc model is used for the following configuration:

- o Create one or more OSE gateways in the serving domain.
- o Create underlying connections between the OSE gateway and other SD-WAN domain gateways, including control plane and data plane.
- o Create overlay tunnels between the OSE gateway and other SD-WAN domain gateways with Tunnel setup parameters, such as IPsec Tunnel related authentication and encryption parameters.
- o Create segment mappings between the OSE gateway and other SD-WAN domain gateways with segment related parameters, such as VLAN ID or VRF ID.

For the configuration of network elements may be done using NETCONF [RFC6241] or any other configuration (or "southbound") interface such as Command Line Interface (CLI) in combination with device-specific and protocol-specific YANG data models.

The usage of this service model is not limited to this example: it can be used by any component of the management system but not directly by network elements.

6. Design of the Data Model

The SD-WAN OSE service model currently has two YANG modules.

6.1. OSE Gateway Service Model

The aim of OSE Gateway module is to define parameters for connection setup between SD-WAN domains. As specified by RFC4364, this model defines Option A and Option B to interconnect the different domain. The option B allows one to minimize configuration inputs and allows the solution to scale really high because only the BGP RIBs store all the inter-AS / inter-SD-WAN VPN routes. MP-BGP can run a single label stack within the GRE tunnel, between the NNI nodes such that the MPLS label will be used for traffic segmentation. In the cases, where L3VPN Inter-AS Option B is not supported, revert to MP-BGP based Inter-AS VPN Option A while using MPLS labels. The option A

requires Orchestration layer to signal underlying SD-WAN domains to configure and instantiate VRF instances per tenant, as well as MP-BGP based L3VPN configuration and instantiation per tenant. This option can still run on GRE or IPSec tunnels while providing isolation from underlay changes and dependencies and MPLS label within the GRE tunnel will provide per tenant service level separation.

- o ose-gateway: Gateway name and Gateway ID are specified for each domain.
- o tunnel: describes encap-type in the interconnection points, and authentication and encryption are also specified to secure the interconnection between SD-WAN domains.
- o ose-interworking-option: MP-BGP based L3VPN Inter-AS Option B with MPLS labels and Inter-AS Option A are defined.
- o ose-gateway control plane peering: Control Plane peering between SD-WAN Edge Nodes which exchanges routes and additional reachability information as well as forward transit traffic. For good HA and resiliency characteristics, it is recommended to establish control plane sessions between each node.
- o segment: to guarantee end to end secure traffic, the segment traffic from a specific domain needs to cross connect to the target segment through an OSE gateway.

The complete data hierarchy is presented as follows:

```

module: ietf-ose-gateway-svc
+--rw ose-gateways
  +--rw ose-gateway* [gw-id]
    +--rw gw-id          uint32
    +--rw gw-name?       string
    +--rw peer-list* [name]
      +--rw name          string
      +--rw peer-gw-id?   uint32
      +--rw peer-gw-name? string
      +--rw ose-interworking-option? enumeration
      +--rw encap-type?   enumeration
      +--rw auth-type?    enumeration
      +--rw crypto-option? enumeration
    +--rw segment-list* [segment-name]
      +--rw segment-name  string
      +--rw vlan-id?      uint16 {ose-option-a}?
      +--rw vrf-id?       uint16 {ose-option-b}?
      +--rw segment-type? enumeration
      +--rw crossconnects* [ccname]
        +--rw ccname      string
        +--rw gateway-reference?
          | -> ../../../../peer-list/peer-gw-id
        +--rw peer-seg-name? string
        +--rw peer-seg-id-vlan? uint16 {ose-option-a}?
        +--rw peer-seg-id-vrf?  uint16 {ose-option-b}?

```

6.2. OSE Path Service Model

Path management module defines automatic path selection policy for traffic across the domain. Policy control will take shape in the form of an ordered list. Each item in the list will be evaluated to match the traffic classifier. The first match will result in processing the matched traffic according to the associated link & path policy. In turn, the link & path policy will be framed in the context of the Performance SLA associated to the links and paths.

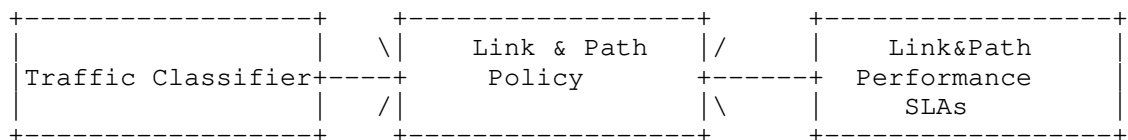


figure 2

Traffic classification rules are handled by the "traffic-class" container. The traffic-classification-policy container is an ordered list of rules that match a flow or application and set the

appropriate business-priority and make link or path selection. This business priority can be factored into the path selection decision.

The client orchestrator can define the match using an application reference or a flow definition that is more specific (e.g., based on Layer 3 source and destination addresses, Layer 4 ports, and Layer 4 protocol).

The link or path selection is defined as a list of services properties. Describes the policy for how links should be selected for the specified traffic flow. The properties are as follows:

- o mode: Describes the policy for how links should be selected for the specified traffic flow. Values are: 1-Automatic 2-Primary/preferred 3-Lowest cost
- o physical-port: describe the WAN port number
- o service-type: Commodity refers to broadband Internet links; Wireless refers to subset of 3G/4G/LTE and upcoming 5G; Private refers to private circuits such as Ethernet, T1, etc
- o service-provider: Specifies the name of provider per enumerated list of providers globally.
- o path-selection-mode: Describes the policy for how paths should be selected for the specified traffic flow. This includes the policy option for portions of traffic to not be sent across the SD-WAN overlay tunnel. Values are: 1 - "Drop" 2 - "UnderNon overlay" 3 - "Overlay".

A custom SLA profile is defined as a list of services properties. The properties are as follows:

- o delay: defines the latency constraint of a specific traffic class.
- o jitter: defines the jitter constraint of a specific traffic class.
- o loss: defines the loss constraint of a specific traffic class.

The complete data hierarchy is presented as follows:

```

module: ietf-ose-path-svc
+--rw path-svc
|   +--rw path-policy* [policy-name]
|   |   +--rw policy-name          string
|   |   +--rw flow-classification* [class-name]
|   |   |   +--rw class-name      string
|   |   |   +--rw dscp?          inet:dscp
|   |   |   +--rw ipv4-src-prefix? inet:ipv4-prefix
|   |   |   +--rw ipv6-src-prefix? inet:ipv6-prefix
|   |   |   +--rw ipv4-dst-prefix? inet:ipv4-prefix
|   |   |   +--rw ipv6-dst-prefix? inet:ipv6-prefix
|   |   |   +--rw l4-src-port?    inet:port-number
|   |   |   +--rw l4-src-port-range
|   |   |   |   +--rw lower-port?  inet:port-number
|   |   |   |   +--rw upper-port?  inet:port-number
|   |   |   +--rw l4-dst-port?    inet:port-number
|   |   |   +--rw l4-dst-port-range
|   |   |   |   +--rw lower-port?  inet:port-number
|   |   |   |   +--rw upper-port?  inet:port-number
|   |   |   +--rw protocol-field? union
|   |   +--rw application-classification* [application-class-name]
|   |   |   +--rw application-class-name string
|   |   |   +--rw category-id?          uint32
|   |   |   +--rw application-id?       uint32
|   |   +--rw user* [list-name]
|   |   |   +--rw list-name      string
|   |   |   +--rw user-id*       string
|   |   |   +--rw user-group*    string
|   |   +--rw site*              uint32
|   |   +--rw link-path-policy
|   |   |   +--rw business-priority? enumeration
|   |   |   +--rw link-selection-mode
|   |   |   |   +--rw mode?          enumeration
|   |   |   |   +--rw physical-port? uint32
|   |   |   |   +--rw service-type?  enumeration
|   |   |   |   +--rw service-provider? string
|   |   |   +--rw path-selection-mode? enumeration
|   +--rw traffic-sla
|   |   +--rw traffic-sla* [custom_sla_name]
|   |   |   +--rw custom_sla_name  string
|   |   |   +--rw direction?       identityref
|   |   |   +--rw latency?         uint32
|   |   |   +--rw jitter?          uint32
|   |   |   +--rw packet-loss-rate? uint32

```

7. SD-WAN OSE Gateway Service YANG Module

```
<CODE BEGINS> file "ietf-ose-gateway-svc@2019-06-10.yang"
module ietf-ose-gateway-svc {
  namespace "urn:ietf:params:xml:ns:yang:ietf-ose-gateway-svc";
  prefix ose-gw-svc;

  organization
    "IETF foo Working Group.";
  contact
    "WG List: foo@ietf.org
     Editor: ";
  description
    "The YANG module defines a generic service configuration
     model for interworking between different SD-WAN domains.";

  revision 2019-06-10 {
    description
      "Initial revision";
    reference
      "A YANG Data Model for SD-WAN service configuration of
       gateway-svc.";
  }

  feature ose-option-a {
    description
      "This feature means that ose reachability service option-A is
       supported by the Serving SDWAN manager";
    reference "ONUG-OSE-2 SDWAN Reachability and Segmentation
              Specification";
  }

  feature ose-option-b {
    description
      "This feature means that ose reachability service option-B
       is supported by the Serving SDWAN manager";
    reference "ONUG-OSE-2 SDWAN Reachability and Segmentation
              Specification";
  }

  container ose-gateways {
    list ose-gateway {
      key "gw-id";
      leaf gw-id {
        type uint32;
        description
          "Identifier for Gateway.";
      }
    }
  }
}
```

```
leaf gw-name {
  type string;
  description
    "OSE gateway name.";
}
list peer-list {
  key "name";
  leaf name {
    type string;
    description
      "Peer Name.";
  }
  leaf peer-gw-id {
    type uint32;
    description
      "Identifier for the remote peer gateway.";
  }
  leaf peer-gw-name {
    type string;
    description
      "Name of remote peer gateway. ";
  }
}
leaf ose-interworking-option {
  type enumeration {
    enum ose-option-a {
      description
        "MP-BGP based Inter-AS VPN Option A with MPLS labels.";
    }
    enum ose-option-b {
      description
        "MP-BGP based L3VPN Inter-AS Option B with MPLS
        labels.";
    }
  }
  default "ose-option-b";
  description
    "OSE Gateway interworking options.";
}
leaf encap-type {
  type enumeration {
    enum ipsec_tunnel {
      description
        "The encapsulation option is IPSec Tunnel mode per
        RFC4303.";
    }
    enum ipsec_transport {
      description
        "The encapsulation option is IPSec Transport mode
```

```
        per RFC4303.";
    }
    enum gre {
        description
            "The encapsulation option is GRE tunnel per.";
    }
}
description
    "The encapsulation options of OSE Gateway interworking.";
}
leaf auth-type {
    type enumeration {
        enum psk {
            description
                "Pre-Shared Key (PSK).";
        }
        enum pki {
            description
                "Public Key Infrastructure.";
        }
    }
}
description
    "authentication type.";
}
leaf crypto-option {
    type enumeration {
        enum aes-128 {
            description
                "crypto algorithm.";
        }
        enum aes-256 {
            description
                "crypto algorithm.";
        }
        enum aes-256-gcm {
            description
                "crypto algorithm.";
        }
    }
}
description
    "Crypto algorithm selection. Others to be added.";
}
description
    "OSE Gateway peer gateway list.";
}
list segment-list {
    key "segment-name";
    leaf segment-name {
```



```
        type string;
        description
            "segment name.";
    }
    leaf vlan-id {
        if-feature "ose-option-a";
        type uint16;
        description
            "vlan ID.";
    }
    leaf vrf-id {
        if-feature "ose-option-b";
        type uint16;
        description
            "vrf ID.";
    }
    leaf segment-type {
        type enumeration {
            enum overlay {
                description
                    "overlay NNI interworking.";
            }
            enum nsw {
                description
                    "underlay NNI interworking.";
            }
        }
        description
            "segment type.";
    }
    list crossconnects {
        key "ccname";
        leaf ccname {
            type string;
            description
                "cross connection name.";
        }
        leaf gateway-reference {
            type leafref {
                path "../../peer-list/peer-gw-id";
            }
            description
                "Specify the OSE gateway to be cross-connected
                with the segment.";
        }
        leaf peer-seg-name {
            type string;
            description
```

```
        "Peer segment name.";
    }
    leaf peer-seg-id-vlan {
        if-feature "ose-option-a";
        type uint16;
        description
            "Peer segment vlan ID.";
    }
    leaf peer-seg-id-vrf {
        if-feature "ose-option-b";
        type uint16;
        description
            "Peer Segment vrf ID.";
    }
    description
        "Cross connection List";
}
description
    "Segment List";
}
description
    "OSE gateway list.";
}
description
    "OSE gateway container.";
}
}
```

<CODE ENDS>

8. SD-WAN OSE Path Service YANG Module

```
<CODE BEGINS> file "ietf-ose-path-svc@2019-06-10.yang"
module ietf-ose-path-svc {
    namespace "urn:ietf:params:xml:ns:yang:ietf-ose-path-svc";
    prefix ose-path-svc;

    import ietf-inet-types {
        prefix inet;
    }

    organization
        "IETF foo Working Group.";
    contact
        "WG List: foo@ietf.org
        Editor: ";
    description
        "The YANG module defines a generic service configuration
```

```
    model for interworking between different SD-WAN domains.";

revision 2019-06-10 {
  description
    "Initial revision";
  reference
    "A YANG Data Model for SD-WAN service configuration of
    path-svc.";
}

identity traffic-direction {
  description
    "Base identity for traffic direction.";
}

identity upstream {
  base traffic-direction;
  description
    "Identity for Site-to-WAN direction.";
}

identity downstream {
  base traffic-direction;
  description
    "Identity for WAN-to-Site direction.";
}

identity both {
  base traffic-direction;
  description
    "Identity for both WAN-to-Site direction
    and Site-to-WAN direction.";
}

identity protocol-type {
  description
    "Base identity for protocol field type.";
}

identity tcp {
  base protocol-type;
  description
    "TCP protocol type.";
}

identity udp {
  base protocol-type;
  description
```

```
    "UDP protocol type.";
}

identity icmp {
    base protocol-type;
    description
        "ICMP protocol type.";
}

identity icmp6 {
    base protocol-type;
    description
        "ICMPv6 protocol type.";
}

identity gre {
    base protocol-type;
    description
        "GRE protocol type.";
}

identity ipip {
    base protocol-type;
    description
        "IP-in-IP protocol type.";
}

identity hop-by-hop {
    base protocol-type;
    description
        "Hop-by-Hop IPv6 header type.";
}

identity routing {
    base protocol-type;
    description
        "Routing IPv6 header type.";
}

identity esp {
    base protocol-type;
    description
        "ESP header type.";
}

identity ah {
    base protocol-type;
    description
```

```
    "AH header type.";
}

container path-svc {
  description
    "Container for application aware path selection policy.";
  list path-policy {
    key "policy-name";
    description
      "List for path selection policy.";
    leaf policy-name {
      type string;
      description
        "Policy name.";
    }
  }
  list flow-classification {
    key "class-name";
    description
      "List for traffic classification.";
    leaf class-name {
      type string;
      description
        "Traffic classification name.";
    }
    leaf dscp {
      type inet:dscp;
      description
        "DSCP value.";
    }
    leaf ipv4-src-prefix {
      type inet:ipv4-prefix;
      description
        "Match on IPv4 src address.";
    }
    leaf ipv6-src-prefix {
      type inet:ipv6-prefix;
      description
        "Match on IPv6 src address.";
    }
    leaf ipv4-dst-prefix {
      type inet:ipv4-prefix;
      description
        "Match on IPv4 dst address.";
    }
    leaf ipv6-dst-prefix {
      type inet:ipv6-prefix;
      description
        "Match on IPv6 dst address.";
    }
  }
}
```

```
}
leaf l4-src-port {
  type inet:port-number;
  must 'current() < ../l4-src-port-range/lower-port or '+
  'current() > ../l4-src-port-range/upper-port' {
    description
      "If l4-src-port and l4-src-port-range/lower-port and
      upper-port are set at the same time, l4-src-port
      should not overlap with l4-src-port-range.";
  }
  description
    "Match on Layer 4 src port.";
}
container l4-src-port-range {
  leaf lower-port {
    type inet:port-number;
    description
      "Lower boundary for port.";
  }
  leaf upper-port {
    type inet:port-number;
    must '. >= ../lower-port' {
      description
        "Upper boundary for port. If it
        exists, the upper boundary must be
        higher than the lower boundary.";
    }
    description
      "Upper boundary for port.";
  }
  description
    "Match on Layer 4 src port range. When
    only the lower-port is present, it represents
    a single port. When both the lower-port and
    upper-port are specified, it implies
    a range inclusive of both values.";
}
leaf l4-dst-port {
  type inet:port-number;
  must 'current() < ../l4-dst-port-range/lower-port or '+
  'current() > ../l4-dst-port-range/upper-port' {
    description
      "If l4-dst-port and l4-dst-port-range/lower-port
      and upper-port are set at the same time,
      l4-dst-port should not overlap with
      l4-src-port-range.";
  }
  description
```

```
        "Match on Layer 4 dst port.";
    }
    container l4-dst-port-range {
        leaf lower-port {
            type inet:port-number;
            description
                "Lower boundary for port.";
        }
        leaf upper-port {
            type inet:port-number;
            must '.. >= ../lower-port' {
                description
                    "Upper boundary must be
                     higher than lower boundary.";
            }
            description
                "Upper boundary for port. If it exists,
                 upper boundary must be higher than lower
                 boundary.";
        }
        description
            "Match on Layer 4 dst port range. When only
             lower-port is present, it represents a single
             port. When both lower-port and upper-port are
             specified, it implies a range inclusive of both
             values.";
    }
    leaf protocol-field {
        type union {
            type uint8;
            type identityref {
                base protocol-type;
            }
        }
        description
            "Match on IPv4 protocol or IPv6 Next Header field.";
    }
}
list application-classification {
    key "application-class-name";
    description
        "List for application.";
    leaf application-class-name {
        type string;
        description
            "Application classification name.";
    }
    leaf category-id {
```

```
        type uint32;
        description
            "Describe the application category, e.g. Media,
             Peer2Peer.";
    }
    leaf application-id {
        type uint32;
        description
            "Describe the application and sub-application flows
             as well.";
    }
}
list user {
    key "list-name";
    description
        "List for User.";
    leaf list-name {
        type string;
        description
            "User list name.";
    }
    leaf-list user-id {
        type string;
        description
            "User list.";
    }
    leaf-list user-group {
        type string;
        description
            "User group list.";
    }
}
leaf-list site {
    type uint32;
    description
        "Describe the enterprise site or set of sites.";
}
container link-path-policy {
    description
        "Container for path selection policy.";
    leaf business-priority {
        type enumeration {
            enum high {
                description
                    "Refers to high priority.";
            }
            enum normal {
                description

```



```
        "Refers to normal priority.";
    }
    enum low {
        description
            "Refers to low priority..";
    }
    enum voice {
        description
            "Refers to voice priority.";
    }
    enum critical_data {
        description
            "Refers to critical_data priority.";
    }
    enum transactional {
        description
            "Refers to transactional priority.";
    }
    enum user-defined {
        description
            "Refers to user-defined priority.";
    }
}
description
    "Describes the business priority for the matched traffic or
    application.";
}
container link-selection-mode {
    description
        "Describes the policy for how links should be selected for
        the specified traffic flow.";
    leaf mode {
        type enumeration {
            enum automatic {
                description
                    "Refers to automatic mode with all the WAN link
                    service.";
            }
            enum primary {
                description
                    "For certain traffic requiring high security or to
                    use a limited usage based circuit.";
            }
            enum lowest-cost {
                description
                    "For certain traffic only low cost WAN link could
                    be used.";
            }
        }
    }
}
```

```
    }
    description
        "Automatic option needs to take the SLA profile into
        consideration; Primary and lowest-cost are NOT
        automatic.";
    }
    leaf physical-port {
        type uint32;
        description
            "When in NOT automatic mode, specify the physical-port.";
    }
    leaf service-type {
        type enumeration {
            enum commodity {
                description
                    "Refers to broadband Internet links.";
            }
            enum wireless {
                description
                    "Refers to subset of 3G/4G/LTE and upcoming 5G.";
            }
            enum private {
                description
                    "Refers to private circuits such as Ethernet, T1,
                    etc.";
            }
        }
        description
            "When in NOT automatic mode, specify the physical-port,
            service-type.";
    }
    leaf service-provider {
        type string;
        description
            "When in NOT automatic mode, specify the name of
            provider.";
    }
}
leaf path-selection-mode {
    type enumeration {
        enum drop {
            description
                "Specify to drop the traffic.";
        }
        enum underlay {
            description
                "Specify the underlay path.";
        }
    }
}
```

```
        enum overlay {
            description
                "Specify the overlay path.";
        }
    }
    default "overlay";
    description
        "Describes the policy for how paths should be selected for
        the specified traffic flow. If a destination for a traffic
        flow can be reached through both the overlay as well as
        the underlay, specify a preference .";
    }
}
}
container traffic-sla {
    description
        "Container for traffic SLA measurement.";
    list traffic-sla {
        key "custom_sla_name";
        description
            "List for traffic sla profile";
        leaf custom_sla_name {
            type string;
            description
                "customer traffic sla name";
        }
        leaf direction {
            type identityref {
                base traffic-direction;
            }
            default "both";
            description
                "The direction to which the QoS profile
                is applied: upstream or downstream.";
        }
        leaf latency {
            type uint32;
            units "msec";
            description
                "Downstream or upstream latency observed on the path in msec";
        }
        leaf jitter {
            type uint32;
            units "msec";
            description
                "Jitter observed on the path in msec";
        }
    }
}
```

```
    leaf packet-loss-rate {  
      type uint32 {  
        range "0..100";  
      }  
      units "percent";  
      description  
        "Percentage of packet loss observed on the path for the  
        upstream and downstream";  
    }  
  }  
}  
}
```

<CODE ENDS>

9. 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:

- o /ose-path/service

The entries in the list above include the whole ose path service configurations which the customer subscribes, and indirectly create or modify the path selection configurations. Unexpected changes to these entries could lead to service disruption and/or network misbehavior.

- o /ose-gateways/ose-gateway

The entries in the list above include the whole ose gateway service configurations which the customer subscribes, and indirectly create or modify the PE,ASBR device configurations. Unexpected changes to these entries could lead to service disruption and/or network misbehavior.

- o /ose-gateways/ose-gateway/peer-list

The entries in the list above include the peer list configurations. As above, unexpected changes to these entries could lead to service disruption and/or network misbehavior.

- o /ose-gateways/ose-gateway/segment-list

The entries in the list above include the segment list configurations. As above, unexpected changes to these entries could lead to service disruption and/or network misbehavior.

10. IANA Considerations

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

URI: urn:ietf:params:xml:ns:yang:ietf-ose-path-svc
Registrant Contact: The IESG
XML: N/A; the requested URI is an XML namespace.

URI: urn:ietf:params:xml:ns:yang:ietf-ose-gateway-svc
Registrant Contact: The IESG
XML: N/A; the requested URI is an XML namespace.

This document registers two YANG modules in the YANG Module Names registry [RFC6020].

Name: ietf-ose-path-svc
Namespace: urn:ietf:params:xml:ns:yang:ietf-ose-path-svc
Prefix: path-svc
Reference: RFC xxxx
Name: ietf-ose-gateway-svc
Namespace: urn:ietf:params:xml:ns:yang:ietf-ose-gateway-svc
Prefix: reach-vpn
Reference: RFC xxxx

11. References

11.1. 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>>.
- [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>>.

11.2. Informative References

- [I-D.rosen-bess-secure-l3vpn] Rosen, E. and R. Bonica, "Augmenting RFC 4364 Technology to Provide Secure Layer L3VPNs over Public Infrastructure", draft-rosen-bess-secure-l3vpn-01 (work in progress), June 2018.
- [ONUG] Group, O. S. W., Ed., "ONUG Software-Defined WAN Use Case: A white paper from the ONUG SD-WAN Working Group", October 2014.
- [OSE] Group, O. O. W., Ed., "ONUG SOFTWARE DEFINED WAN (SD-WAN): NETWORK ARCHITECTURE FRAMEWORK".
- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <<https://www.rfc-editor.org/info/rfc3688>>.
- [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>>.
- [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>>.
- [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>>.

- [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>>.
- [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>>.
- [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>>.

Appendix A. Acknowledges

Authors' Addresses

Steve Wood (editor)
Cisco Systems

Email: swood1@cisco.com

Bo Wu (editor)
Huawei Technologies, Co., Ltd

Email: lane.wubo@huawei.com

Qin Wu (editor)
Huawei Technologies, Co., Ltd

Email: bill.wu@huawei.com

Conrad Menezes
HPE Aruba

Email: conrad.menezes@hpe.com

Networking Working Group
Internet-Draft
Intended status: Informational
Expires: April 28, 2020

Q. Wu, Ed.
Huawei
M. Boucadair, Ed.
Orange
D. Lopez
Telefonica I+D
C. Xie
China Telecom
L. Geng
China Mobile
October 26, 2019

A Framework for Automating Service and Network Management with YANG
draft-wu-model-driven-management-virtualization-07

Abstract

Data models for service and network management provides a programmatic approach for representing (virtual) services or networks and deriving (1) configuration information that will be communicated to network and service components that are used to build and deliver the service and (2) state information that will be monitored and tracked. Indeed, data models can be used during various phases of the service and network management life cycle, such as service instantiation, service provisioning, optimization, monitoring, and diagnostic. Also, data models are instrumental in the automation of network management. They also provide closed-loop control for the sake of adaptive and deterministic service creation, delivery, and maintenance.

This document provides a framework that describes and discusses an architecture for service and network management automation that takes advantage of YANG modeling technologies. This framework is drawn from a network provider perspective irrespective of the origin of a data module; it can accommodate even modules that are developed outside the IETF.

The document aims to exemplify an approach that specifies the journey from technology-agnostic services to technology-specific actions.

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 April 28, 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	5
3. Architectural Concepts & Goals	5
3.1. Data Models: Layering and Representation	5
3.2. Automation of Service Delivery Procedures	6
3.3. Service Fullfillment Automation	7
3.4. YANG Modules Integration	7
4. Architecture Overview	8
4.1. Service Lifecycle Management Procedure	9
4.1.1. Service Exposure	10
4.1.2. Service Creation/Modification	10
4.1.3. Service Optimization	10
4.1.4. Service Diagnosis	11
4.1.5. Service Decommission	11
4.2. Service Fullfillment Management Procedure	11
4.2.1. Intended Configuration Provision	11
4.2.2. Configuration Validation	12
4.2.3. Performance Monitoring	12
4.2.4. Fault Diagnostic	13
4.3. Multi-layer/Multi-domain Service Mapping	13
4.4. Service Decomposing	13

5. YANG Data Model Integration Examples	13
5.1. L3VPN Service Delivery	13
5.2. VN Lifecycle Management Example	15
6. Security Considerations	16
7. IANA Considerations	16
8. Acknowledgements	16
9. Contributors	16
10. Informative References	17
Appendix A. Layered YANG Modules Example Overview	25
A.1. Service Models: Definition and Samples	25
A.2. Network Models: Definitions and Samples	26
A.3. Device Models: Definitions and Samples	29
A.3.1. Model Composition	30
A.3.2. Device Models: Definitions and Samples	30
Authors' Addresses	33

1. Introduction

The service management system usually comprises service activation/provision and service operation. Current service delivery procedures, from the processing of customer's requirements and order to service delivery and operation, typically assume the manipulation of data sequentially into multiple OSS/BSS applications that may be managed by different departments within the service provider's organization (e.g., billing factory, design factory, network operation center, etc.). In addition, many of these applications have been developed in-house over the years and operating in a silo mode:

- o The lack of standard data input/output (i.e., data model) also raises many challenges in system integration and often results in manual configuration tasks.
- o Secondly, many current service fulfillment system might have limited visibility to the network and therefore have slow response to the network changes.

Software Defined Networking (SDN) becomes crucial to address these challenges. SDN techniques [RFC7149] are meant to automate the overall service delivery procedures and typically rely upon (standard) data models that are used to not only reflect service providers' savoir-faire but also to dynamically instantiate and enforce a set of (service-inferred) policies that best accommodate what has been (contractually) defined (and possibly negotiated) with the customer. [RFC7149] provides a first tentative to rationalize that service provider's view on the SDN space by identifying concrete technical domains that need to be considered and for which solutions can be provided:

- o Techniques for the dynamic discovery of topology, devices, and capabilities, along with relevant information and data models that are meant to precisely document such topology, devices, and their capabilities.
- o Techniques for exposing network services [RFC8309] and their characteristics.
- o Techniques used by service-requirement-derived dynamic resource allocation and policy enforcement schemes, so that networks can be programmed accordingly.
- o Dynamic feedback mechanisms that are meant to assess how efficiently a given policy (or a set thereof) is enforced from a service fulfillment and assurance perspective.

Models are key for each of these technical items. Service and network management automation is an important step to improve the agility of network operations and infrastructures. Models are also important to ease integrating multi-vendor solutions.

YANG module developers have taken both top-down and bottom-up approaches to develop modules [RFC8199], and also to establish a mapping between network technology and customer requirements on the top or abstracting common construct from various network technologies on the bottom. At the time of writing this document (2019), there are many data models including configuration and service models that have been specified or are being specified by the IETF. They cover many of the networking protocols and techniques. However, how these models work together to configure a device, manage a set of devices involved in a service, or even provide a service is something that is not currently documented either within the IETF or other SDOs (e.g., MEF).

This document provides a framework that describes and discusses an architecture for service and network management automation that takes advantage of YANG modeling technologies and investigates how different layer YANG data models interact with each other (e.g., service mapping, model composing) in the context of service delivery and fulfillment.

This framework is drawn from a network provider perspective irrespective of the origin of a data module; it can accommodate even modules that are developed outside the IETF.

The document also identifies a list of use cases to exemplify the proposed approach, but it does not claim to be exhaustive.

2. Terminology

The following terms are defined in [RFC8309][RFC8199] and are not redefined here:

- o Network Operator
- o Customer
- o Service
- o Data Model
- o Service Model
- o Network Element Module

The document makes use of the following terms:

Network Model: The Network Model describes network level abstraction or various aspects of a network infrastructure, including devices and their subsystems, and relevant protocols operating at the link and network layers across multiple devices. It can be used by a network operator to allocate the resource(e.g., tunnel resource, topology resource) for the service or schedule the resource to meet the service requirements define in the Service Model.

Device Model: Network Element YANG data module described in [RFC8199].

3. Architectural Concepts & Goals

3.1. Data Models: Layering and Representation

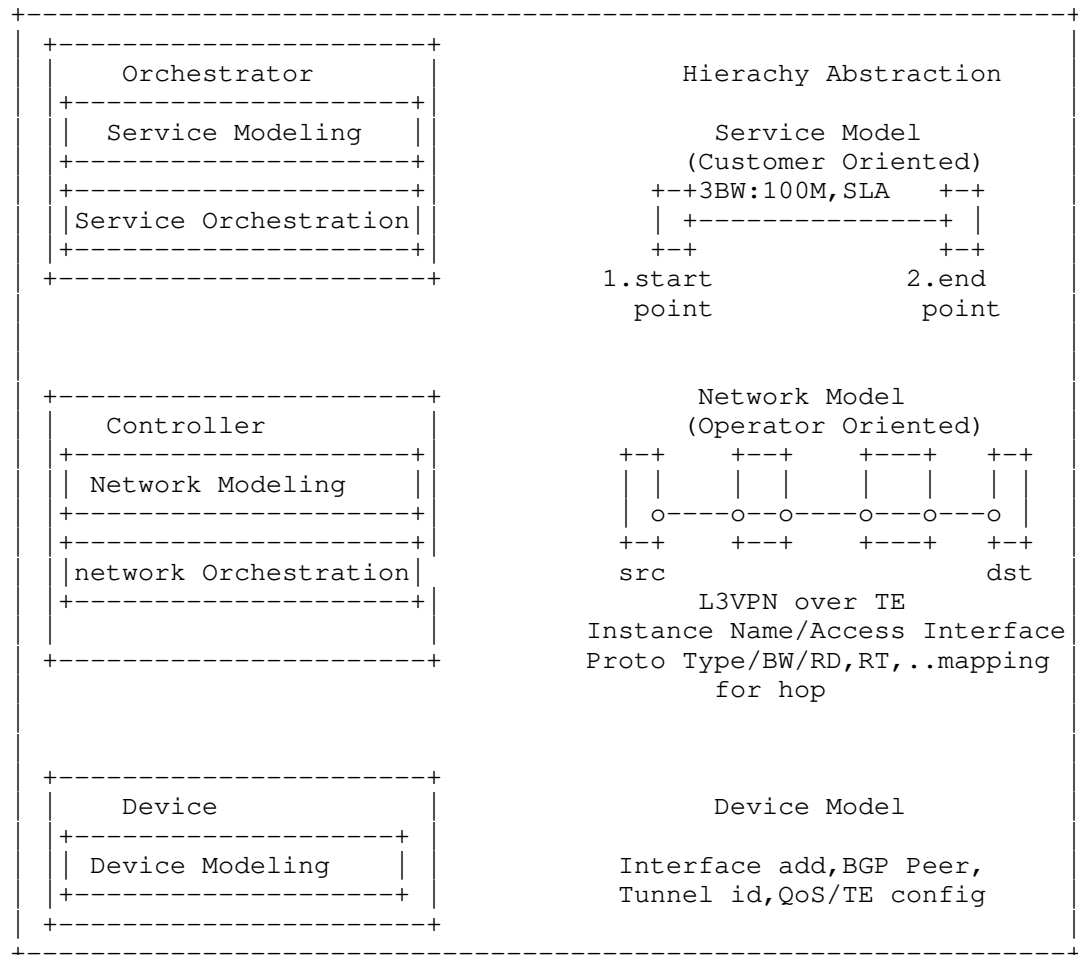
As described in [RFC8199], layering of modules allows for better reusability of lower-layer modules by higher-level modules while limiting duplication of features across layers.

The data modules developed by IETF can be classified into service level, network level and device level modules. Different service model at service level may rely on the same set of network level or device level models. Service models usually follow top down approach and are mostly customer-facing modules providing a common model construct for higher level network services, which can be further mapped to network technology-specific modules at lower layer.

Network level modules are mainly network resource-facing modules and describe various aspects of a network infrastructure, including

devices and their subsystems, and relevant protocols operating at the link and network layers across multiple devices (e.g., Network topology and TE Tunnel modules).

Device level modules usually follow a bottom-up approach and are mostly technology-specific modules used to realize a service.



Layering and representation

3.2. Automation of Service Delivery Procedures

To dynamically offer and deliver service offerings, Service level modules can be used by an operator. One or more monolithic Service modules can be used in the context of a composite service activation

request (e.g., delivery of a caching infrastructure over a VPN). Such modules are used to feed a decision-making intelligence to adequately accommodate customer's needs.

Also, such modules may be used jointly with services that require dynamic invocation. An example is provided by the service modules defined by the DOTS WG to dynamically trigger requests to handle DDoS attacks [I-D.ietf-dots-signal-channel][I-D.ietf-dots-data-channel].

Network level modules can be derived from service level modules and used to provision, monitor, instantiate the service, and provide lifecycle management of network resources (e.g., expose network resources to customers or operators to provide service fulfillment and assurance and allow customers or operators to dynamically adjust the network resources based on service requirements as described in service level modules and the current network performance information described in the telemetry modules).

3.3. Service Fullfillment Automation

To operate the service, Device level modules derived from Service level modules or Network level modules can be used to provision each involved network function/device with the proper configuration information, and operate the network based on service requirements as described in the Service level module(s).

In addition, the operational state including configuration that is in effect together with statistics should be exposed to upper layers to provide better network visibility (and assess to what extent the derived low level modules are consistent with the upper level inputs).

Note that it is important to correlate telemetry data with configuration data to be used for closed loops at the different stages of service delivery, from resource allocation to service operation, in particular.

3.4. YANG Modules Integration

To support top-down service delivery, YANG modules at different level or at the same level need to be integrated together to enable function, feature in the network device and get network setup. For example, the service parameters captured in service level modules need to be decomposed into a set of (configuration/notification) parameters that may be specific to one or more technologies; these technology-specific parameters are grouped together to define technology-specific device level models or network level models.

In addition, these technology-specific device level models or network level models can be further integrated with each other using schema mount mechanism [RFC8528] to provision each involved network function/device or each involved administrative domain to support newly added module or features. A collection of device models integrated together can be loaded and validated during implementation time.

Policies provide a higher layer of abstraction. Policy models can be defined at service level, network level, or device level to provide policy-based management and telemetry automation, e.g., telemetry data can trigger a new policy that captures new network service requirements.

Performance measurement telemetry can be used to provide service assurance at service level or at the network level. Performance measurement telemetry model can tie with network level model or service level model to monitor network performance or service level agreement.

4. Architecture Overview

The architectural considerations described in Section 3 lead to the architecture described in this section and illustrated in Figure 1.

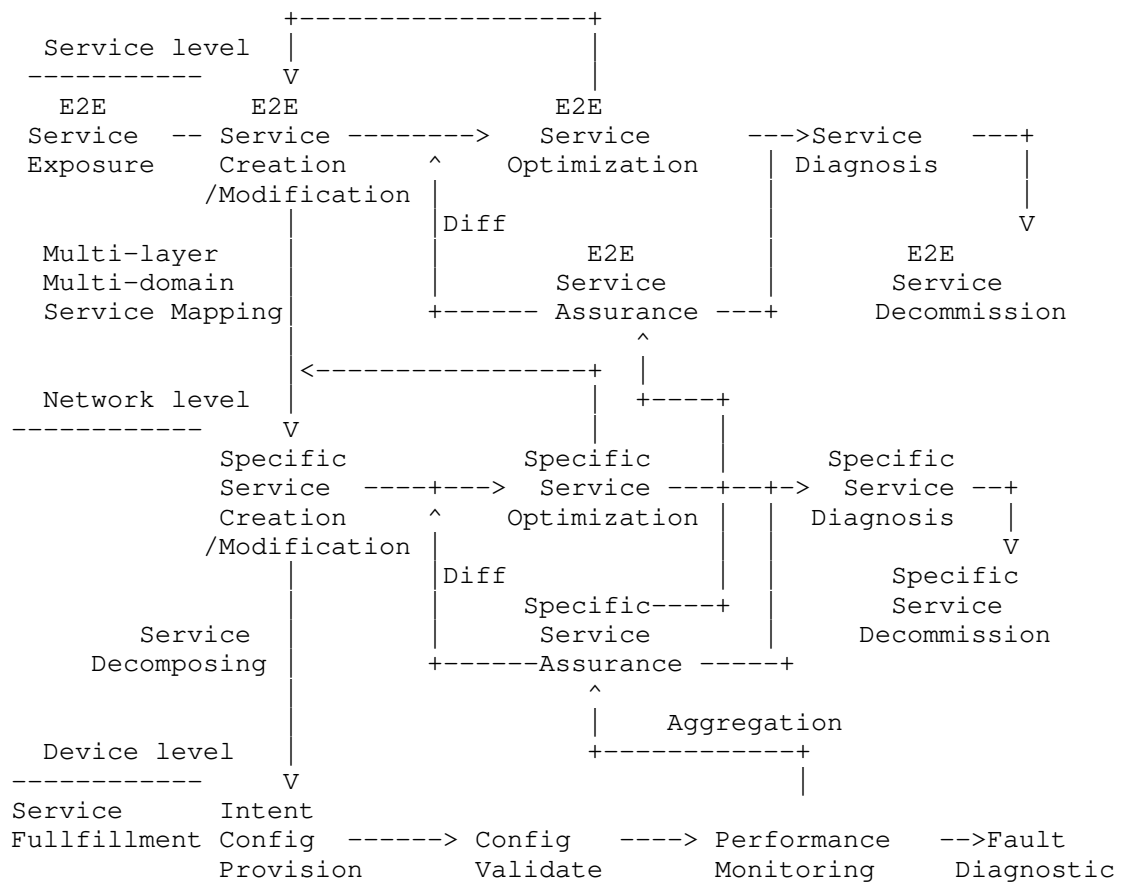


Figure 1: Service and Network Lifecycle Management

4.1. Service Lifecycle Management Procedure

Service lifecycle management includes end to end service lifecycle management at the service level and specific network lifecycle management at the network level. The end-to-end service lifecycle management is multi-domain or multi-layer service management while specific service lifecycle management is domain specific or layer specific service lifecycle management.

- o Note: Clarify what is meant by "domain".

4.1.1.1. Service Exposure

A service in the context of this document (sometimes called a Network Service) is some form of connectivity between customer sites and the Internet or between customer sites across the network operator's network and across the Internet.

Service exposure is used to capture services offered to customers (ordering and order handling). One typical example is that a customer can use a L3SM service model to request L3VPN service by providing the abstract technical characterization of the intended service between customer sites.

Service model catalogs can be created along to expose the various services and the information needed to invoke/order a given service.

4.1.1.2. Service Creation/Modification

A customer is (usually) unaware of the technology that the network operator has available to deliver the service, so the customer does not make requests specific to the underlying technology but is limited to making requests specific to the service that is to be delivered. This service request can be issued using the service model.

The service orchestrator/management system maps such service request to its view. This view can be described as a network model and this mapping may include a choice of which networks and technologies to use depending on which service features have been requested.

In addition, a customer may require to change underlying network infrastructure to adapt to new customer's needs and service requirements. This service modification can be issued in the same service model used by the service request.

4.1.1.3. Service Optimization

Service optimization is a technique that gets the configuration of the network updated due to network change, incident mitigation, or new service requirements. One typical example is once the tunnel or the VPN is setup, Performance monitoring information or telemetry information per tunnel or per VPN can be collected and fed into the management system, if the network performance doesn't meet the service requirements, the management system can create new VPN policies capturing network service requirements and populate them into the network.

Both network performance information and policies can be modelled using YANG. With Policy-based management, self-configuration and self-optimization behavior can be specified and implemented.

4.1.4. Service Diagnosis

Operations, Administration, and Maintenance (OAM) are important networking functions for service diagnosis that allow operators to:

- o monitor network communications (i.e., reachability verification and Continuity Check)
- o troubleshoot failures (i.e., fault verification and localization)
- o monitor service-level agreements and performance (i.e., performance management)

When the network is down, service diagnosis should be in place to pinpoint the problem and provide recommendation (or instructions) for the network recovery.

The service diagnosis information can be modelled as technology-independent RPC operations for OAM protocols and technology-independent abstraction of key OAM constructs for OAM protocols [RFC8531][RFC8533]. These models can provide consistent configuration, reporting, and presentation for the OAM mechanisms used to manage the network.

4.1.5. Service Decommission

Service decommission allow the customer to stop the service and remove the service from active status and release the network resource that is allocated to the service. Customer can also use the service model to withdraw the subscription to a service.

4.2. Service Fullfillment Management Procedure

4.2.1. Intended Configuration Provision

Intended configuration at the device level is derived from network model at the network level or service model at the service level and represents the configuration that the system attempts to apply. Take L3SM service model as an example, to deliver a L3VPN service, we need to map L3VPN service view defined in Service model into detailed intended configuration view defined by specific configuration models for network elements, configuration information includes:

- o VRF definition, including VPN Policy expression

- o Physical Interface
- o IP layer (IPv4, IPv6)
- o QoS features such as classification, profiles, etc.
- o Routing protocols: support of configuration of all protocols listed in the document, as well as routing policies associated with those protocols.
- o Multicast Support
- o NAT or address sharing
- o Security function

This specific configuration models can be used to configure PE and CE devices within the site, e.g., A BGP policy model can be used to establish VPN membership between sites and VPN Service Topology.

4.2.2. Configuration Validation

Configuration validation is used to validate intended configuration and ensure the configuration take effect. For example, a customer creates an interface "et-0/0/0" but the interface does not physically exist at this point, then configuration data appears in the <intended> status but does not appear in <operational> datastore.

4.2.3. Performance Monitoring

When configuration is in effect in the device, <operational> datastore holds the complete operational state of the device including learned, system, default configuraton and system state. However the configurations and state of a particular device does not have the visibility to the whole network or information of the flow packets are going to take through the entire network. Therefore it becomes more difficult to operate the network without understanding the current status of the network.

The management system should subscribe to updates of a YANG datastore in all the network devices for performance monitoring purpose and build full topological visibility to the network by aggregating and filtering these operational state from different sources.

4.2.4. Fault Diagnostic

When configuration is in effect in the device, some device may be misconfigured(e.g.,device links are not consistent on both sides of the network connection), network resources be misallocated and services may be negatively affected without knowing what is going on in the network.

Technology-dependent nodes and remote procedure call (RPC) commands are defined in technology-specific YANG data models which can use and extend the base model described in Section 4.1.4can be used to deal with these challenges.

These RPC command recieved in the technology dependent node can be used to trigger technology specific OAM message exchange for fault verification and fault isolation,e.g., TRILL Multicast Tree Verification (MTV) RPC command [I-D.ietf-trill-yang-oam] can be used to trigger Multi-Destination Tree Verification Message defined in [RFC7455] to verify TRILL distribution tree integrity.

4.3. Multi-layer/Multi-domain Service Mapping

Multi-layer/Multi-domain Service Mapping allow you map end to end abstract view of the service segmented at different layer or different administrative domain into domain specific view. One example is to map service parameters in L3VPN service model into configuration parameters such as RD, RT, and VRF in L3VPN network model. Another example is to map service parameters in L3VPN service model into TE tunnel parameter (e.g.,Tunnel ID) in TE model and VN parameters (e.g., AP list, VN member) in TEAS VN model [I-D.ietf-teas-actn-vn-yang].

4.4. Service Decomposing

Service Decomposing allows to decompose service model at the service level or network model at the network level into a set of device/function models at the device level. These device models may be tied to specific device type or classified into a collection of related YANG modules based on service type and feature offered and load at the implementation time before configuration is loaded and validated.

5. YANG Data Model Integration Examples

5.1. L3VPN Service Delivery

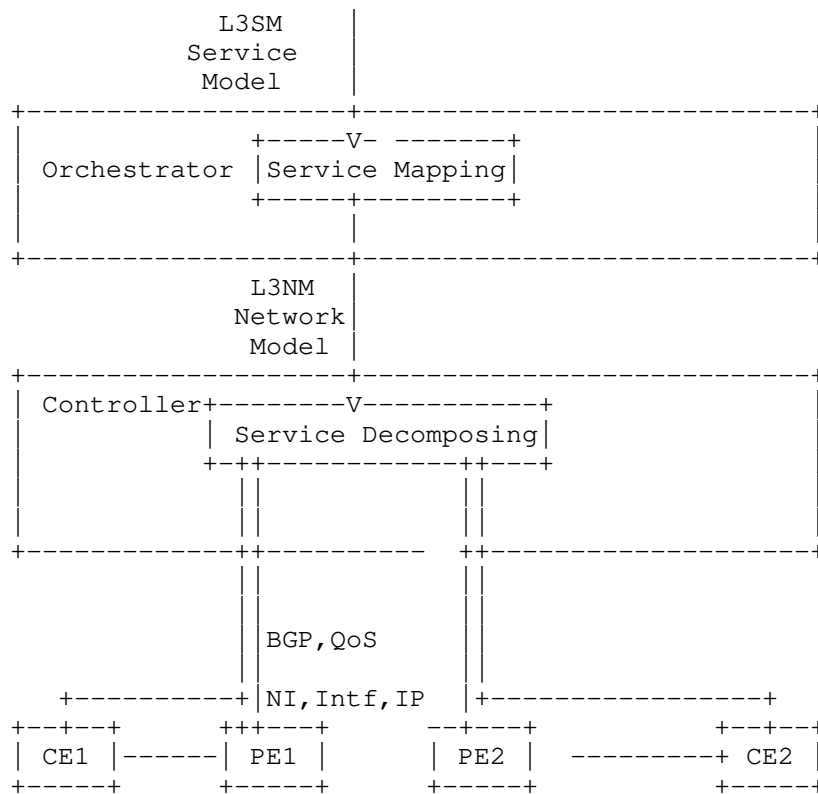


Figure 2: L3VPN Service Delivery Example

In reference to Figure 2, the following steps are performed to deliver the L3VPN service within the network management automation architecture defined in this document:

1. Customer Requests to create two sites based on L3SM Service model with each having one network access connectivity:

Site A: Network-Access A, Bandwidth=20M, for class "foo",
guaranteed-bw-percent = 10, One-Way-Delay=70 msec

Site B: Network-Access B, Bandwidth=30M, for class "foo1",
guaranteed-bw-percent = 15, One-Way-Delay=60 msec

2. The Orchestrator extracts the service parameters from the L3SM model. Then, it uses them as input to translate them into an orchestrated configuration of network elements (e.g., RD, RT, VRF, etc.) that is part of the L3NM network model.

3. The Controller takes orchestrated configuration parameters in the L3NM network model and translates them into orchestrated configuration of network elements that is part of BGP model, QoS model, Network Instance model, IP management model, interface model, etc.

5.2. VN Lifecycle Management Example

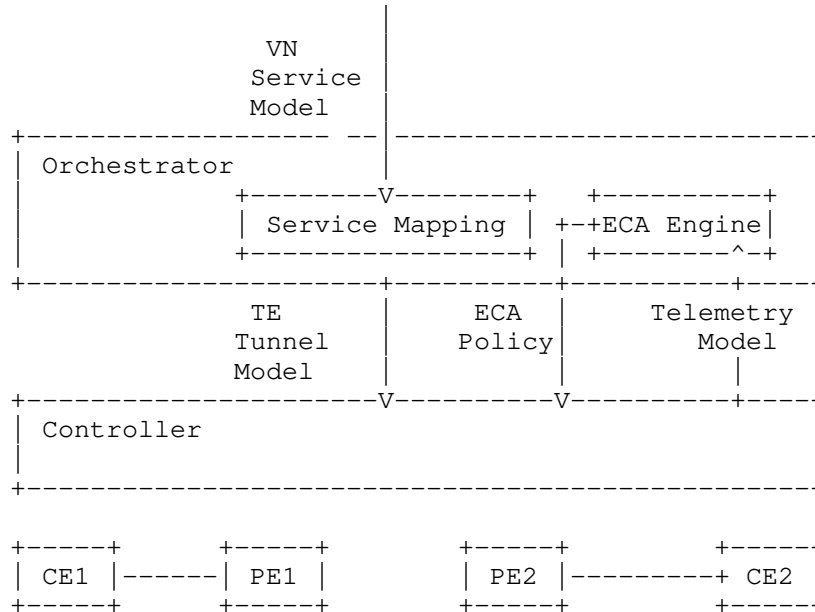


Figure 3

In reference to Figure 3, the following steps are performed to deliver the VN service within the network management automation architecture defined in this document:

1. Customer requests to create 'VN' based on Access point, association between VN and Access point, VN member defined in the VN YANG module.
2. The orchestrator creates the single abstract node topology based on the information captured in an VN YANG module.
3. The Customer exchanges connectivity-matrix on abstract node and explicit path using TE topology model with the orchestrator. This information can be used to instantiate VN and setup tunnels between source and destination endpoints.

4. The telemetry which augments the TEAS VN model and corresponding TE Tunnel model can be used to notify all the parameter changes and network performance change related to VN topology or Tunnel [I-D.ietf-teas-actn-pm-telemetry-autonomics]. This information can be further used as input to ECA engine in the orchestrator and generate ECA policy model to optimize the network.

6. Security Considerations

Security considerations specific to each of the technologies and protocols listed in the document are discussed in the specification documents of each of these techniques.

(Potential) security considerations specific to this document are listed below:

- o Create forwarding loops by mis-configuring the underlying network.
- o Leak sensitive information: special care should be considered when translating between the various layers introduced in the document.
- o ...tbc

7. IANA Considerations

There are no IANA requests or assignments included in this document.

8. Acknowledgements

Thanks to Joe Clark, Greg Mirsky, and Shunsuke Homma for the review.

9. Contributors

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

Luis Miguel Contreras Murillo
Telifonica

Email: luismiguel.contrerasmurillo@telefonica.com

Oscar Gonzalez de Dios
Telefonica
Madrid
ES

Email: oscar.gonzalezdedios@telefonica.com

Chongfeng Xie
China Telecom
Beijing
China

Email: xiechf.bri@chinatelecom.cn

Weiqiang Cheng
China Mobile

Email: chengweiqiang@chinamobile.com

Young Lee
Sung Kyun Kwan University

Email: younglee.tx@gmail.com

10. Informative References

[I-D.arkko-arch-virtualization]

Arkko, J., Tantsura, J., Halpern, J., and B. Varga,
"Considerations on Network Virtualization and Slicing",
draft-arkko-arch-virtualization-01 (work in progress),
March 2018.

[I-D.asechoud-netmod-diffserv-model]

Choudhary, A., Shah, S., Jethanandani, M., Liu, B., and N.
Strahle, "YANG Model for Diffserv", draft-asechoud-netmod-
diffserv-model-03 (work in progress), June 2015.

- [I-D.claccla-netmod-model-catalog]
Clarke, J. and B. Claise, "YANG module for yangcatalog.org", draft-claccla-netmod-model-catalog-03 (work in progress), April 2018.
- [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-01 (work in progress), July 2019.
- [I-D.ietf-bess-evpn-yang]
Brissette, P., Shah, H., Hussain, I., Tiruveedhula, K., and J. Rabadan, "Yang Data Model for EVPN", draft-ietf-bess-evpn-yang-07 (work in progress), March 2019.
- [I-D.ietf-bess-l2vpn-yang]
Shah, H., Brissette, P., Chen, I., Hussain, I., Wen, B., and K. Tiruveedhula, "YANG Data Model for MPLS-based L2VPN", draft-ietf-bess-l2vpn-yang-10 (work in progress), July 2019.
- [I-D.ietf-bess-l3vpn-yang]
Jain, D., Patel, K., Brissette, P., Li, Z., Zhuang, S., Liu, X., Haas, J., Esale, S., and B. Wen, "Yang Data Model for BGP/MPLS L3 VPNs", draft-ietf-bess-l3vpn-yang-04 (work in progress), October 2018.
- [I-D.ietf-bfd-yang]
Rahman, R., Zheng, L., Jethanandani, M., Networks, J., and G. Mirsky, "YANG Data Model for Bidirectional Forwarding Detection (BFD)", draft-ietf-bfd-yang-17 (work in progress), August 2018.
- [I-D.ietf-ccamp-alarm-module]
Vallin, S. and M. Bjorklund, "YANG Alarm Module", draft-ietf-ccamp-alarm-module-09 (work in progress), April 2019.
- [I-D.ietf-ccamp-flexigrid-media-channel-yang]
Madrid, U., Perdices, D., Lopezalvarez, V., Dios, O., King, D., Lee, Y., and G. Galimberti, "YANG data model for Flexi-Grid media-channels", draft-ietf-ccamp-flexigrid-media-channel-yang-02 (work in progress), March 2019.

- [I-D.ietf-ccamp-flexigrid-yang]
Madrid, U., Perdices, D., Lopezalvarez, V., King, D., and Y. Lee, "YANG data model for Flexi-Grid Optical Networks", draft-ietf-ccamp-flexigrid-yang-04 (work in progress), July 2019.
- [I-D.ietf-ccamp-llcsm-yang]
Lee, Y., Lee, K., Zheng, H., Dhody, D., Dios, O., and D. Ceccarelli, "A YANG Data Model for L1 Connectivity Service Model (L1CSM)", draft-ietf-ccamp-llcsm-yang-10 (work in progress), September 2019.
- [I-D.ietf-ccamp-mw-yang]
Ahlberg, J., Ye, M., Li, X., Spreafico, D., and M. Vaupotic, "A YANG Data Model for Microwave Radio Link", draft-ietf-ccamp-mw-yang-13 (work in progress), November 2018.
- [I-D.ietf-ccamp-otn-topo-yang]
Zheng, H., Guo, A., Busi, I., Sharma, A., Liu, X., Belotti, S., Xu, Y., Wang, L., and O. Dios, "A YANG Data Model for Optical Transport Network Topology", draft-ietf-ccamp-otn-topo-yang-08 (work in progress), September 2019.
- [I-D.ietf-ccamp-otn-tunnel-model]
Zheng, H., Busi, I., Belotti, S., Lopezalvarez, V., and Y. Xu, "OTN Tunnel YANG Model", draft-ietf-ccamp-otn-tunnel-model-08 (work in progress), October 2019.
- [I-D.ietf-ccamp-wson-tunnel-model]
Lee, Y., Zheng, H., Guo, A., Lopezalvarez, V., King, D., Yoon, B., and R. Vilata, "A Yang Data Model for WSON Tunnel", draft-ietf-ccamp-wson-tunnel-model-04 (work in progress), September 2019.
- [I-D.ietf-dots-data-channel]
Boucadair, M. and R. K, "Distributed Denial-of-Service Open Threat Signaling (DOTS) Data Channel Specification", draft-ietf-dots-data-channel-31 (work in progress), July 2019.
- [I-D.ietf-dots-signal-channel]
K, R., Boucadair, M., Patil, P., Mortensen, A., and N. Teague, "Distributed Denial-of-Service Open Threat Signaling (DOTS) Signal Channel Specification", draft-ietf-dots-signal-channel-37 (work in progress), July 2019.

- [I-D.ietf-idr-bgp-model]
Jethanandani, M., Patel, K., Hares, S., and J. Haas, "BGP YANG Model for Service Provider Networks", draft-ietf-idr-bgp-model-07 (work in progress), October 2019.
- [I-D.ietf-ippm-stamp-yang]
Mirsky, G., Xiao, M., and W. Luo, "Simple Two-way Active Measurement Protocol (STAMP) Data Model", draft-ietf-ippm-stamp-yang-05 (work in progress), October 2019.
- [I-D.ietf-ippm-twamp-yang]
Civil, R., Morton, A., Rahman, R., Jethanandani, M., and K. Pentikousis, "Two-Way Active Measurement Protocol (TWAMP) Data Model", draft-ietf-ippm-twamp-yang-13 (work in progress), July 2018.
- [I-D.ietf-mpls-base-yang]
Saad, T., Raza, K., Gandhi, R., Liu, X., and V. Beeram, "A YANG Data Model for MPLS Base", draft-ietf-mpls-base-yang-11 (work in progress), September 2019.
- [I-D.ietf-pim-igmp-ml-d-snooping-yang]
Zhao, H., Liu, X., Liu, Y., Sivakumar, M., and A. Peter, "A Yang Data Model for IGMP and MLD Snooping", draft-ietf-pim-igmp-ml-d-snooping-yang-08 (work in progress), June 2019.
- [I-D.ietf-pim-igmp-ml-d-yang]
Liu, X., Guo, F., Sivakumar, M., McAllister, P., and A. Peter, "A YANG Data Model for Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD)", draft-ietf-pim-igmp-ml-d-yang-15 (work in progress), June 2019.
- [I-D.ietf-pim-yang]
Liu, X., McAllister, P., Peter, A., Sivakumar, M., Liu, Y., and f. hu, "A YANG Data Model for Protocol Independent Multicast (PIM)", draft-ietf-pim-yang-17 (work in progress), May 2018.
- [I-D.ietf-rtgwg-device-model]
Lindem, A., Berger, L., Bogdanovic, D., and C. Hopps, "Network Device YANG Logical Organization", draft-ietf-rtgwg-device-model-02 (work in progress), March 2017.

- [I-D.ietf-rtgwg-policy-model]
Qu, Y., Tantsura, J., Lindem, A., and X. Liu, "A YANG Data Model for Routing Policy Management", draft-ietf-rtgwg-policy-model-07 (work in progress), September 2019.
- [I-D.ietf-software-iftunnel]
Boucadair, M., Farrer, I., and R. Asati, "Tunnel Interface Types YANG Module", draft-ietf-software-iftunnel-07 (work in progress), June 2019.
- [I-D.ietf-software-yang]
Farrer, I. and M. Boucadair, "YANG Modules for IPv4-in-IPv6 Address plus Port (A+P) Softwires", draft-ietf-software-yang-16 (work in progress), January 2019.
- [I-D.ietf-spring-sr-yang]
Litkowski, S., Qu, Y., Lindem, A., Sarkar, P., and J. Tantsura, "YANG Data Model for Segment Routing", draft-ietf-spring-sr-yang-13 (work in progress), July 2019.
- [I-D.ietf-supra-generic-policy-data-model]
Halpern, J. and J. Strassner, "Generic Policy Data Model for Simplified Use of Policy Abstractions (SUPA)", draft-ietf-supra-generic-policy-data-model-04 (work in progress), June 2017.
- [I-D.ietf-teas-actn-vn-yang]
Lee, Y., Dhody, D., Ceccarelli, D., Bryskin, I., and B. Yoon, "A Yang Data Model for VN Operation", draft-ietf-teas-actn-vn-yang-06 (work in progress), July 2019.
- [I-D.ietf-teas-sf-aware-topo-model]
Bryskin, I., Liu, X., Lee, Y., Guichard, J., Contreras, L., Ceccarelli, D., and J. Tantsura, "SF Aware TE Topology YANG Model", draft-ietf-teas-sf-aware-topo-model-03 (work in progress), March 2019.
- [I-D.ietf-teas-te-service-mapping-yang]
Lee, Y., Dhody, D., Fioccola, G., WU, Q., Ceccarelli, D., and J. Tantsura, "Traffic Engineering (TE) and Service Mapping Yang Model", draft-ietf-teas-te-service-mapping-yang-02 (work in progress), September 2019.
- [I-D.ietf-teas-yang-l3-te-topo]
Liu, X., Bryskin, I., Beeram, V., Saad, T., Shah, H., and O. Dios, "YANG Data Model for Layer 3 TE Topologies", draft-ietf-teas-yang-l3-te-topo-05 (work in progress), July 2019.

- [I-D.ietf-teas-yang-path-computation]
Busi, I. and S. Belotti, "Yang model for requesting Path Computation", draft-ietf-teas-yang-path-computation-06 (work in progress), July 2019.
- [I-D.ietf-teas-yang-rsvp-te]
Beeram, V., Saad, T., Gandhi, R., Liu, X., Bryskin, I., and H. Shah, "A YANG Data Model for RSVP-TE Protocol", draft-ietf-teas-yang-rsvp-te-07 (work in progress), July 2019.
- [I-D.ietf-teas-yang-sr-te-topo]
Liu, X., Bryskin, I., Beeram, V., Saad, T., Shah, H., and S. Litkowski, "YANG Data Model for SR and SR TE Topologies", draft-ietf-teas-yang-sr-te-topo-05 (work in progress), July 2019.
- [I-D.ietf-teas-yang-te]
Saad, T., Gandhi, R., Liu, X., Beeram, V., and I. Bryskin, "A YANG Data Model for Traffic Engineering Tunnels and Interfaces", draft-ietf-teas-yang-te-21 (work in progress), April 2019.
- [I-D.ietf-teas-yang-te-topo]
Liu, X., Bryskin, I., Beeram, V., Saad, T., Shah, H., and O. Dios, "YANG Data Model for Traffic Engineering (TE) Topologies", draft-ietf-teas-yang-te-topo-22 (work in progress), June 2019.
- [I-D.ietf-trill-yang-oam]
Kumar, D., Senevirathne, T., Finn, N., Salam, S., Xia, L., and H. Weiguo, "YANG Data Model for TRILL Operations, Administration, and Maintenance (OAM)", draft-ietf-trill-yang-oam-05 (work in progress), March 2017.
- [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>>.
- [RFC4664] Andersson, L., Ed. and E. Rosen, Ed., "Framework for Layer 2 Virtual Private Networks (L2VPNs)", RFC 4664, DOI 10.17487/RFC4664, September 2006, <<https://www.rfc-editor.org/info/rfc4664>>.
- [RFC4761] Kompella, K., Ed. and Y. Rekhter, Ed., "Virtual Private LAN Service (VPLS) Using BGP for Auto-Discovery and Signaling", RFC 4761, DOI 10.17487/RFC4761, January 2007, <<https://www.rfc-editor.org/info/rfc4761>>.

- [RFC4762] Lasserre, M., Ed. and V. Kompella, Ed., "Virtual Private LAN Service (VPLS) Using Label Distribution Protocol (LDP) Signaling", RFC 4762, DOI 10.17487/RFC4762, January 2007, <<https://www.rfc-editor.org/info/rfc4762>>.
- [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>>.
- [RFC7149] Boucadair, M. and C. Jacquenet, "Software-Defined Networking: A Perspective from within a Service Provider Environment", RFC 7149, DOI 10.17487/RFC7149, March 2014, <<https://www.rfc-editor.org/info/rfc7149>>.
- [RFC7276] Mizrahi, T., Sprecher, N., Bellagamba, E., and Y. Weingarten, "An Overview of Operations, Administration, and Maintenance (OAM) Tools", RFC 7276, DOI 10.17487/RFC7276, June 2014, <<https://www.rfc-editor.org/info/rfc7276>>.
- [RFC7297] Boucadair, M., Jacquenet, C., and N. Wang, "IP Connectivity Provisioning Profile (CPP)", RFC 7297, DOI 10.17487/RFC7297, July 2014, <<https://www.rfc-editor.org/info/rfc7297>>.
- [RFC7455] Senevirathne, T., Finn, N., Salam, S., Kumar, D., Eastlake 3rd, D., Aldrin, S., and Y. Li, "Transparent Interconnection of Lots of Links (TRILL): Fault Management", RFC 7455, DOI 10.17487/RFC7455, March 2015, <<https://www.rfc-editor.org/info/rfc7455>>.
- [RFC8077] Martini, L., Ed. and G. Heron, Ed., "Pseudowire Setup and Maintenance Using the Label Distribution Protocol (LDP)", STD 84, RFC 8077, DOI 10.17487/RFC8077, February 2017, <<https://www.rfc-editor.org/info/rfc8077>>.
- [RFC8194] Schoenwaelder, J. and V. Bajpai, "A YANG Data Model for LMAP Measurement Agents", RFC 8194, DOI 10.17487/RFC8194, August 2017, <<https://www.rfc-editor.org/info/rfc8194>>.
- [RFC8199] Bogdanovic, D., Claise, B., and C. Moberg, "YANG Module Classification", RFC 8199, DOI 10.17487/RFC8199, July 2017, <<https://www.rfc-editor.org/info/rfc8199>>.
- [RFC8299] Wu, Q., Ed., Litkowski, S., Tomotaki, L., and K. Ogaki, "YANG Data Model for L3VPN Service Delivery", RFC 8299, DOI 10.17487/RFC8299, January 2018, <<https://www.rfc-editor.org/info/rfc8299>>.

- [RFC8309] Wu, Q., Liu, W., and A. Farrel, "Service Models Explained", RFC 8309, DOI 10.17487/RFC8309, January 2018, <<https://www.rfc-editor.org/info/rfc8309>>.
- [RFC8328] Liu, W., Xie, C., Strassner, J., Karagiannis, G., Klyus, M., Bi, J., Cheng, Y., and D. Zhang, "Policy-Based Management Framework for the Simplified Use of Policy Abstractions (SUPA)", RFC 8328, DOI 10.17487/RFC8328, March 2018, <<https://www.rfc-editor.org/info/rfc8328>>.
- [RFC8345] Clemm, A., Medved, J., Varga, R., Bahadur, N., Ananthakrishnan, H., and X. Liu, "A YANG Data Model for Network Topologies", RFC 8345, DOI 10.17487/RFC8345, March 2018, <<https://www.rfc-editor.org/info/rfc8345>>.
- [RFC8346] Clemm, A., Medved, J., Varga, R., Liu, X., Ananthakrishnan, H., and N. Bahadur, "A YANG Data Model for Layer 3 Topologies", RFC 8346, DOI 10.17487/RFC8346, March 2018, <<https://www.rfc-editor.org/info/rfc8346>>.
- [RFC8349] Lhotka, L., Lindem, A., and Y. Qu, "A YANG Data Model for Routing Management (NMDA Version)", RFC 8349, DOI 10.17487/RFC8349, March 2018, <<https://www.rfc-editor.org/info/rfc8349>>.
- [RFC8466] Wen, B., Fioccola, G., Ed., Xie, C., and L. Jalil, "A YANG Data Model for Layer 2 Virtual Private Network (L2VPN) Service Delivery", RFC 8466, DOI 10.17487/RFC8466, October 2018, <<https://www.rfc-editor.org/info/rfc8466>>.
- [RFC8512] Boucadair, M., Ed., Sivakumar, S., Jacquenet, C., Vinapamula, S., and Q. Wu, "A YANG Module for Network Address Translation (NAT) and Network Prefix Translation (NPT)", RFC 8512, DOI 10.17487/RFC8512, January 2019, <<https://www.rfc-editor.org/info/rfc8512>>.
- [RFC8513] Boucadair, M., Jacquenet, C., and S. Sivakumar, "A YANG Data Model for Dual-Stack Lite (DS-Lite)", RFC 8513, DOI 10.17487/RFC8513, January 2019, <<https://www.rfc-editor.org/info/rfc8513>>.
- [RFC8519] Jethanandani, M., Agarwal, S., Huang, L., and D. Blair, "YANG Data Model for Network Access Control Lists (ACLs)", RFC 8519, DOI 10.17487/RFC8519, March 2019, <<https://www.rfc-editor.org/info/rfc8519>>.

- [RFC8528] Bjorklund, M. and L. Lhotka, "YANG Schema Mount", RFC 8528, DOI 10.17487/RFC8528, March 2019, <<https://www.rfc-editor.org/info/rfc8528>>.
- [RFC8529] Berger, L., Hopps, C., Lindem, A., Bogdanovic, D., and X. Liu, "YANG Data Model for Network Instances", RFC 8529, DOI 10.17487/RFC8529, March 2019, <<https://www.rfc-editor.org/info/rfc8529>>.
- [RFC8530] Berger, L., Hopps, C., Lindem, A., Bogdanovic, D., and X. Liu, "YANG Model for Logical Network Elements", RFC 8530, DOI 10.17487/RFC8530, March 2019, <<https://www.rfc-editor.org/info/rfc8530>>.
- [RFC8531] Kumar, D., Wu, Q., and Z. Wang, "Generic YANG Data Model for Connection-Oriented Operations, Administration, and Maintenance (OAM) Protocols", RFC 8531, DOI 10.17487/RFC8531, April 2019, <<https://www.rfc-editor.org/info/rfc8531>>.
- [RFC8532] Kumar, D., Wang, Z., Wu, Q., Ed., Rahman, R., and S. Raghavan, "Generic YANG Data Model for the Management of Operations, Administration, and Maintenance (OAM) Protocols That Use Connectionless Communications", RFC 8532, DOI 10.17487/RFC8532, April 2019, <<https://www.rfc-editor.org/info/rfc8532>>.
- [RFC8533] Kumar, D., Wang, M., Wu, Q., Ed., Rahman, R., and S. Raghavan, "A YANG Data Model for Retrieval Methods for the Management of Operations, Administration, and Maintenance (OAM) Protocols That Use Connectionless Communications", RFC 8533, DOI 10.17487/RFC8533, April 2019, <<https://www.rfc-editor.org/info/rfc8533>>.

Appendix A. Layered YANG Modules Example Overview

It is not the intent of this document to provide an inventory of tools and mechanisms used in specific network and service management domains; such inventory can be found in documents such as [RFC7276].

A.1. Service Models: Definition and Samples

As described in [RFC8309], the service is "some form of connectivity between customer sites and the Internet and/or between customer sites across the network operator's network and across the Internet". More concretely, an IP connectivity service can be defined as the IP transfer capability characterized by a (Source Nets, Destination Nets, Guarantees, Scope) tuple where "Source Nets" is a group of

unicast IP addresses, "Destination Nets" is a group of IP unicast and/or multicast addresses, and "Guarantees" reflects the guarantees (expressed in terms of Quality Of Service (QoS), performance, and availability, for example) to properly forward traffic to the said "Destination" [RFC7297].

For example:

- o L3SM model [RFC8299] defines the L3VPN service ordered by a customer from a network operator.
- o L2SM model [RFC8466] defines the L2VPN service ordered by a customer from a network operator.
- o VN model [I-D.ietf-teas-actn-vn-yang] provides a YANG data model generally applicable to any mode of Virtual Network (VN) operation.

A.2. Network Models: Definitions and Samples

Figure 4 depicts a set of Network models such as topology models or tunnel models:

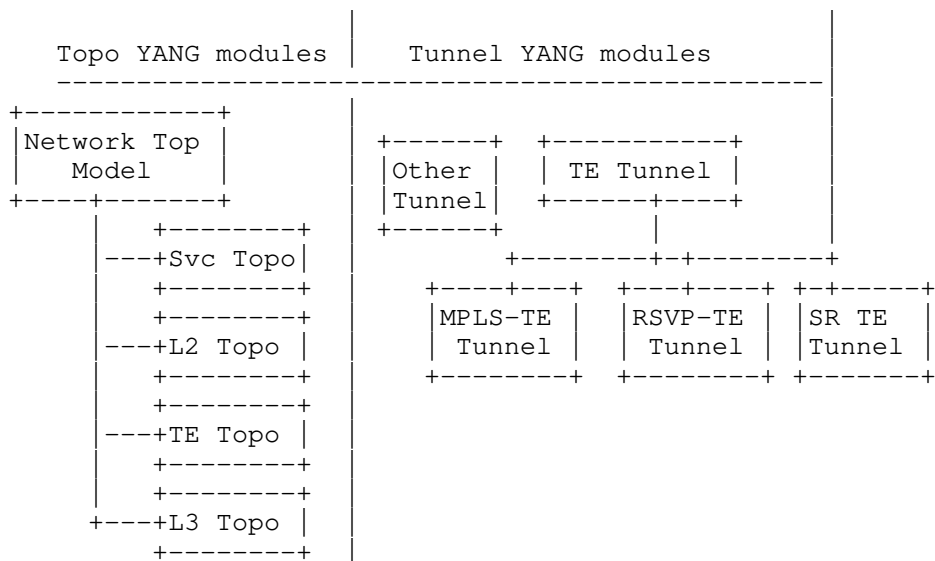


Figure 4: Sample Resource Facing Network Models

Topology YANG module Examples:

- o Network Topology Models: [RFC8345] defines a base model for network topology and inventories. Network topology data include link resource, node resource, and terminate-point resources.
- o TE Topology Models: [I.D-ietf-teas-yang-te-topo] defines a data model for representing and manipulating TE topologies.

This module is extended from network topology model defined in [RFC8345] with TE topologies specifics. This model contains technology-agnostic TE Topology building blocks that can be augmented and used by other technology-specific TE Topology models.

- o L3 Topology Models

[RFC8346] defines a data model for representing and manipulating L3 Topologies. This model is extended from the network topology model defined in [RFC8345] with L3 topologies specifics.

- o L2 Topology Models

[I.D-ietf-i2rs-yang-l2-topology] defines a data model for representing and manipulating L2 Topologies. This model is extended from the network topology model defined in [RFC8345] with L2 topologies specifics.

Tunnel YANG module Examples:

- o Tunnel identities [I-D.ietf-software-iftunnel] to ease manipulating extensions to specific tunnels.
- o TE Tunnel Model

[I.D-ietf-teas-yang-te] defines a YANG module for the configuration and management of TE interfaces, tunnels and LSPs.

- o SR TE Tunnel Model

[I.D-ietf-teas-yang-te] augments the TE generic and MPLS-TE model(s) and defines a YANG module for Segment Routing (SR) TE specific data.

- o MPLS TE Model

[I.D-ietf-teas-yang-te] augments the TE generic and MPLS-TE model(s) and defines a YANG module for MPLS TE configurations, state, RPC and notifications.

- o RSVP-TE MPLS Model

[I.D-ietf-teas-yang-rsvp-te] augments the RSVP-TE generic module with parameters to configure and manage signaling of MPLS RSVP-TE LSPs.

Other Network Models:

- o Path Computation API Model

[I.D-ietf-teas-path-computation] YANG module for a stateless RPC which complements the stateful solution defined in [I.D-ietf-teas-yang-te].

- o OAM Models (including Fault Management (FM) and Performance Monitoring)

[RFC8532] defines a base YANG module for the management of OAM protocols that use Connectionless Communications. [RFC8533] defines a retrieval method YANG module for connectionless OAM protocols. [RFC8531] defines a base YANG module for connection oriented OAM protocols. These three models are intended to provide consistent reporting, configuration and representation for connection-less OAM and Connection oriented OAM separately.

Alarm monitoring is a fundamental part of monitoring the network. Raw alarms from devices do not always tell the status of the network services or necessarily point to the root cause. [I.D-ietf-ccamp-alarm-module] defines a YANG module for alarm management.

- o Generic Policy Model

The Simplified Use of Policy Abstractions (SUPA) policy-based management framework [RFC8328] defines base YANG modules [I-D.ietf-sup-generic-policy-data-model] to encode policy. These models point to device-, technology-, and service-specific YANG modules developed elsewhere. Policy rules within an operator's environment can be used to express high-level, possibly network-wide, policies to a network management function (within a controller, an orchestrator, or a network element). The network management function can then control the configuration and/or monitoring of network elements and services. This document describes the SUPA basic framework, its elements, and interfaces.

A.3. Device Models: Definitions and Samples

Network Element models (Figure 5) are used to describe how a service can be implemented by activating and tweaking a set of functions (enabled in one or multiple devices, or hosted in cloud infrastructures) that are involved in the service delivery. The following figure uses IETF defined models as an example.

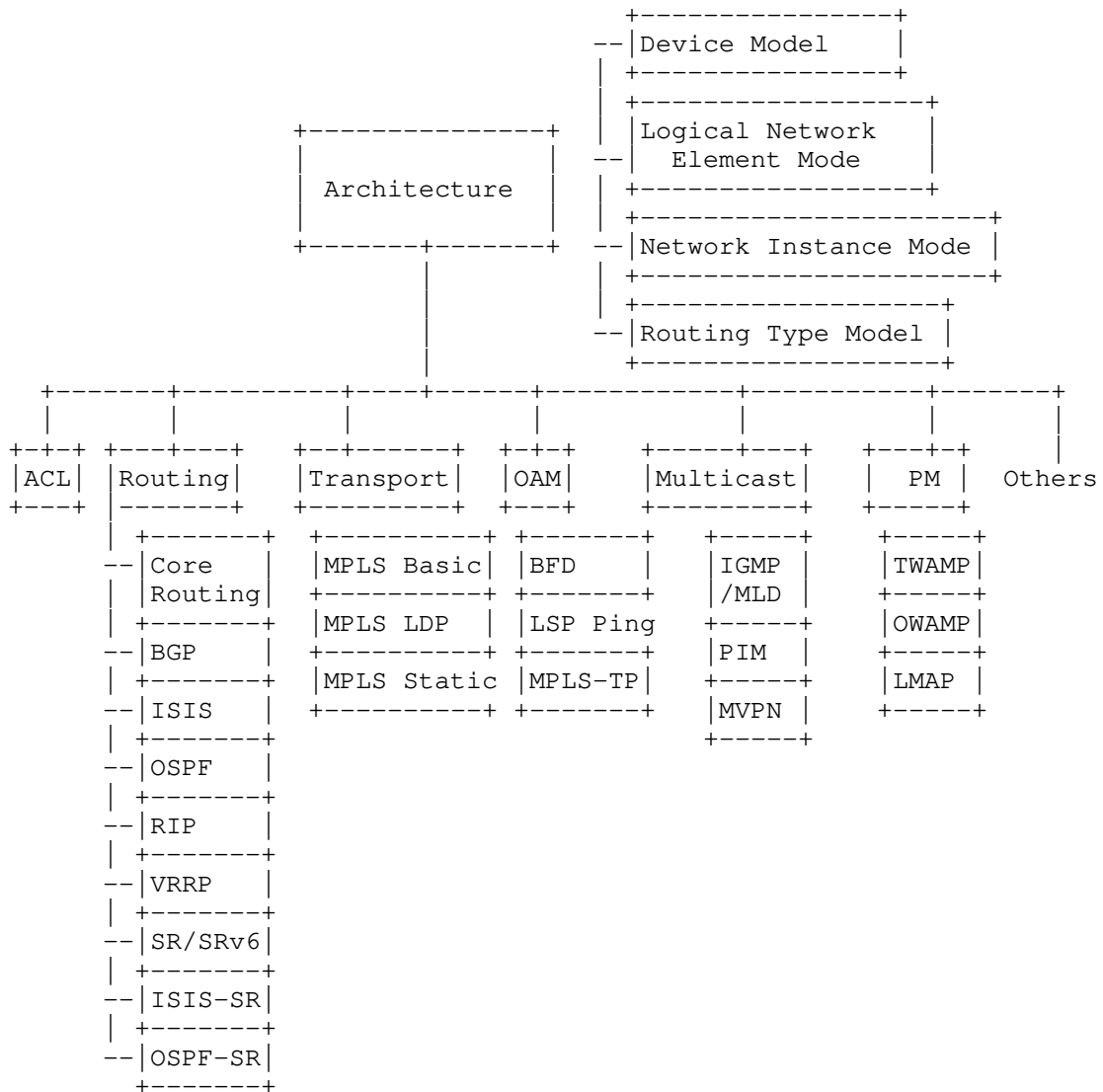


Figure 5: Network Element Modules Overview

A.3.1. Model Composition

- o Device Model

[I.D-ietf-rtgwg-device-model] presents an approach for organizing YANG modules in a comprehensive logical structure that may be used to configure and operate network devices. The structure is itself represented as an example YANG module, with all of the related component models logically organized in a way that is operationally intuitive, but this model is not expected to be implemented.

- o Logical Network Element Model

[RFC8530] defines a logical network element module which can be used to manage the logical resource partitioning that may be present on a network device. Examples of common industry terms for logical resource partitioning are Logical Systems or Logical Routers.

- o Network Instance Model

[RFC8529] defines a network instance module. This module can be used to manage the virtual resource partitioning that may be present on a network device. Examples of common industry terms for virtual resource partitioning are Virtual Routing and Forwarding (VRF) instances and Virtual Switch Instances (VSIs).

A.3.1.1. Schema Mount

Modularity and extensibility were among the leading design principles of the YANG data modeling language. As a result, the same YANG module can be combined with various sets of other modules and thus form a data model that is tailored to meet the requirements of a specific use case. [RFC8528] defines a mechanism, denoted schema mount, that allows for mounting one data model consisting of any number of YANG modules at a specified location of another (parent) schema.

That capability does not cover design time.

A.3.2. Device Models: Definitions and Samples

BGP: [I-D.ietf-idr-bgp-yang-model] defines a YANG module for configuring and managing BGP, including protocol, policy, and operational aspects based on data center, carrier and content provider operational requirements.

- MPLS: [I-D.ietf-mpls-base-yang] defines a base model for MPLS which serves as a base framework for configuring and managing an MPLS switching subsystem. It is expected that other MPLS technology YANG modules (e.g. MPLS LSP Static, LDP or RSVP-TE models) will augment the MPLS base YANG module.
- QoS: [I-D.asechoud-netmod-diffserv-model] describes a YANG module of Differentiated Services for configuration and operations.
- ACL: Access Control List (ACL) is one of the basic elements used to configure device forwarding behavior. It is used in many networking technologies such as Policy Based Routing, Firewalls, etc. [RFC8519] describes a data model of Access Control List (ACL) basic building blocks.
- NAT: For the sake of network automation and the need for programming Network Address Translation (NAT) function in particular, a data model for configuring and managing the NAT is essential. [RFC8512] defines a YANG module for the NAT function covering a variety of NAT flavors such as Network Address Translation from IPv4 to IPv4 (NAT44), Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers (NAT64), customer-side translator (CLAT), Stateless IP/ICMP Translation (SIIT), Explicit Address Mappings (EAM) for SIIT, IPv6-to-IPv6 Network Prefix Translation (NPTv6), and Destination NAT. [RFC8513] specifies a YANG module for the DS-Lite AFTR.
- Stateless Address Sharing: [I-D.ietf-softwire-yang] specifies a YANG module for A+P address sharing, including Lightweight 4over6, Mapping of Address and Port with Encapsulation (MAP-E), and Mapping of Address and Port using Translation (MAP-T) softwire mechanisms.
- Multicast: [I-D.ietf-pim-yang] defines a YANG module that can be used to configure and manage Protocol Independent Multicast (PIM) devices. [I-D.ietf-pim-igmp-mld-yang] defines a YANG module that can be used to configure and manage Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) devices. [I-D.ietf-pim-igmp-mld-snooping-yang] defines a YANG module that can be used to configure and manage Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) Snooping devices.

EVPN: [I-D.ietf-bess-evpn-yang] defines a YANG module for Ethernet VPN services. The model is agnostic of the underlay. It apply to MPLS as well as to VxLAN encapsulation. The model is also agnostic of the services including E-LAN, E-LINE and E-TREE services. This document mainly focuses on EVPN and Ethernet-Segment instance framework.

L3VPN: [I-D.ietf-bess-l3vpn-yang] defines a YANG module that can be used to configure and manage BGP L3VPNs [RFC4364]. It contains VRF specific parameters as well as BGP specific parameters applicable for L3VPNs.

L2VPN: [I-D.ietf-bess-l2vpn-yang] defines a YANG module for MPLS based Layer 2 VPN services (L2VPN) [RFC4664] and includes switching between the local attachment circuits. The L2VPN model covers point-to-point VPWS and Multipoint VPLS services. These services use signaling of Pseudowires across MPLS networks using LDP [RFC8077][RFC4762] or BGP [RFC4761].

Routing Policy: [I-D.ietf-rtgwg-policy-model] defines a YANG module for configuring and managing routing policies in a vendor-neutral way and based on actual operational practice. The model provides a generic policy framework which can be augmented with protocol-specific policy configuration.

BFD: [I-D.ietf-bfd-yang] defines a YANG module that can be used to configure and manage Bidirectional Forwarding Detection (BFD) [RFC5880]. BFD is a network protocol which is used for liveness detection of arbitrary paths between systems.

SR/SRv6: [I-D.ietf-spring-sr-yang] a YANG module for segment routing configuration and operation. [I-D.raza-spring-srv6-yang] defines a YANG module for Segment Routing IPv6 (SRv6) base. The model serves as a base framework for configuring and managing an SRv6 subsystem and expected to be augmented by other SRv6 technology models accordingly.

Core Routing: [RFC8349] defines the core routing data model, which is intended as a basis for future data model development covering more-sophisticated routing systems. It is expected that other Routing technology YANG modules (e.g., VRRP, RIP, ISIS, OSPF models) will augment the Core Routing base YANG module.

PM:

[I.D-ietf-ippm-twamp-yang] defines a data model for client and server implementations of the Two-Way Active Measurement Protocol (TWAMP).

[I.D-ietf-ippm-stamp-yang] defines the data model for implementations of Session-Sender and Session-Reflector for Simple Two-way Active Measurement Protocol (STAMP) mode using YANG.

[RFC8194] defines a data model for Large-Scale Measurement Platforms (LMAPs).

Authors' Addresses

Qin Wu (editor)
Huawei
101 Software Avenue, Yuhua District
Nanjing, Jiangsu 210012
China

Email: bill.wu@huawei.com

Mohamed Boucadair (editor)
Orange
Rennes 35000
France

Email: mohamed.boucadair@orange.com

Diego R. Lopez
Telefonica I+D
Spain

Email: diego.r.lopez@telefonica.com

Chongfeng Xie
China Telecom
Beijing
China

Email: xiechf.bri@chinatelecom.cn

Liang Geng
China Mobile

Email: gengliang@chinamobile.com