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
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:

- 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.

- 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.

- **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.
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.

- 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;

- 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.

```
+--------+----------+
|        |          |
| v      |          |
| +-------------+-------------+ |
| | Data Source: | NPM problem space: |
| | Topology, protocol PDU, | sufficient data type coverage, |
| | RIB, route policy, | sufficient device coverage |
| | statistics... | |
| +-------------+-------------+ |
| | V           | |
| +-------------+-------------+ |
| | Data Generation: | NPM problem space: |
| | data encapsulation, | data model definition, |
| | data serialization, | data process efficiency |
| | data subscription | |
| +-------------+-------------+ |
| | V           | |
| +-------------+-------------+ |
| | Data Transportation: | NPM problem space: |
| | BMP, gRPC, Netconf, | Transportation protocol |
| | BGP-LS, new protocol? | selection, |
| | | exportation efficiency |
| +-------------+-------------+ |
| | V           | |
| +-------------+-------------+ |
| | Data Analysis: | NPM problem space: |
| | Protocol troubleshooting, | data synchronization, |
| | Policy validation, | data parse efficiency |
| | Traffic optimization, | |
| | What-if simulation | |
| +---------------------------+ |
```

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 (caused by import route policy misconfiguration), device clock dis-function with abnormal LSP purge (e.g., 100 times faster) and so on.

- 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.

- The IS-IS neighborship flapping is typically caused by interface flapping, BFD flapping, CPU high and so on. Conventionally, to located 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.

- 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.

- 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 Orginator 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:

- 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.

- Case 2, LSP transmission error, which is typically caused by IS-IS adjacency failure, e.g., link down/BFD down/authentication failure.

- 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 advertise/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:

- 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.

- 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


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


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
Network Slice Provision Models
draft-homma-slice-provision-models-01

Abstract

Network slicing is an approach to provide separate virtual network based on service requirements. It’s a fundamental concept of the 5G, and the architecture and specification is under standardization in several organizations. 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 January 9, 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.  Differentiated Roles in Network Slice Provisioning .......... 3
    1.2.  High-level Problem Statement ................................. 4
2.  Definition of Terms .................................................. 4
3.  General Requirements for Network Slicing .......................... 7
    3.1.  Requirements/Attributes for Network Slice .................... 7
4.  Network Slice Structure ............................................. 8
    4.1.  Resources for Structuring Network Slices ..................... 8
    4.2.  Basic Network Slice Structure ................................ 12
    4.3.  Stakeholders in the Structuring Network Slices ............. 14
5.  Variations of Network Slice Creation .............................. 15
    5.1.  Ready-made Network Slice ...................................... 15
    5.2.  Custom-made Network Slice .................................... 15
    5.3.  semi-Custom-made Network Slice ................................. 15
6.  Network Slice Provision Models ..................................... 15
    6.1.  SaaS-like Model ............................................... 16
        6.1.1.  Capability in SaaS-like Model ............................ 16
    6.2.  PaaS-like Model ................................................ 16
        6.2.1.  Capability in PaaS-like Model ............................ 17
    6.3.  IaaS-like Model ................................................ 17
        6.3.1.  Capability in IaaS-like Model ........................... 17
7.  Security Considerations ............................................. 19

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 applicabilities of existing technologies/solutions to network slicing.

1.1. Differentiated 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.

1.2. 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], and [TS.28.801-3GPP]), 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.

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 NSISs. 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(TRN)-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.

RAN-NSSI: Regardless of RAN deployment (e.g. distributed-RAN, Centralized-RAN or Cloud-RAN), a RAN-NSSI creates a dedicate 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(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 an 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.

3.1. Requirements/Attributes for Network Slice

NS tenants will have specific requirements for network slices depending on the usages or service characteristics. Such requirements or the associated attributes are broken down into concrete design including network topology and configurations of infrastructure resources, and NS is established based on the design. The requirements or attributes on NSs are listed below:

- Requirements/Attributes of Network Resource
  * bandwidth
* latency
* jitter
* packet loss rate
* reliability (e.g., MTBF, MTTF)

Requirements/Attributes of Functionalities Resources

* function type (e.g., security, parental control)
* throughput
* packet error rate
* availability

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 resource 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
  + 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)
- 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
    - Fixed IP address allocation
    - Dynamic IP address allocation
    - The number of registered devices
  - DNS
  - VoIP (SBC, SIP server)
  - Mobile CP function (Ref. 3GPP 5GS)
    - AMF (Access and Mobility management Function)
    - SMF (Session Management Function)
    - PCF (Policy Control Function)
    - UDM (Unified Data Management)
    - NEF (Network Exposure Function)

* Provided VNF Type (e.g., open source, product of vendor#A, ..., 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:

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

---

4.3. Stakeholders in the Structuring Network Slices

Potential stakeholders in network slicing are described below:

- NSSI provider: infrastructure operator
- Intermediate-NSI provider: infrastructure operator, VNO
- E2E-NSI provider: infrastructure operator, VNO, service provider
- NS tenant: infrastructure operator, VNO, service provider, enterprise, mass user
- 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 section classifies NS provision models into three categories defined from aspect that granularity of information exposed to tenants. 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. The capabilities which NS tenants can have on management of NSs would vary depending on the selected provision model.
6.1. 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.

6.1.1. Capability in SaaS-like Model

In SaaS-like Model, an NS is represented for a tenant with attributes values listed in Section 3.1. 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 configuring 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.

6.2. 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.
6.2.1. 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 their 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 the whole of NSI.

6.3. 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.

6.3.1. 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.

6.4. 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.

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-nfvg-slice-gateway]
Homma, S., Foy, X., and A. Galis, "Gateway Function for Network Slicing",
draft-homma-nfvg-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>.
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]).
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
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
SRv6 Path Egress Protection
draft-hu-rtgwg-srv6-egress-protection-02

Abstract
This document describes protocol extensions and procedures for protecting the egress node of a Segment Routing for IPv6 (SRv6) path.

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 January 9, 2020.

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

Fast protection of a transit node of a Segment Routing (SR) path is described in [I-D.bashandy-rtgwg-segment-routing-ti-lfa] and [I-D.hu-spring-segment-routing-proxy-forwarding]. However, these documents do not discuss the procedures for fast protection of the egress node of a Segment Routing for IPv6 (SRv6) path.

[RFC8400] describes the fast protection of egress node(s) of an MPLS TE LSP tunnel including P2P TE LSP tunnel and P2MP TE LSP tunnel in details.

This document specifies protocol extensions and procedures for fast protection of the egress node of an SRv6 path. Egress node and egress as well as fast protection and protection will be used exchangeably.
2. Terminologies

The following terminologies are used in this document.

SR: Segment Routing
SRv6: SR for IPv6
SRH: Segment Routing Header
SID: Segment Identifier
LSP: Label Switched Path
TE: Traffic Engineering
P2MP: Point-to-MultiPoint
P2P: Point-to-Point
CE: Customer Edge
PE: Provider Edge
LFA: Loop-Free Alternate
TI-LFA: Topology Independent LFA
BFD: Bidirectional Forwarding Detection
VPN: Virtual Private Network
L3VPN: Layer 3 VPN
VRF: Virtual Routing and Forwarding
FIB: Forwarding Information Base
PLR: Point of Local Repair
BGP: Border Gateway Protocol
IGP: Interior Gateway Protocol
OSPF: Open Shortest Path First
IS-IS: Intermediate System to Intermediate System
3. SR Path Egress Protection

Figure 1 shows an example of protecting egress PE3 of a SR path, which is from ingress PE1 to egress PE3.

[Diagram]

Node P1’s pre-computed TI-LFA backup path for PE3 is from P1 to PE4 via P2. In normal operations, after receiving a packet with destination PE3, P1 forwards the packet to PE3 according to its FIB. When PE3 receives the packet, it sends the packet to CE2.

When PE3 fails, P1 detects the failure through BFD and forwards the packet to PE4 via the backup path. When PE4 receives the packet, it sends the packet to the same CE2.

In Figure 1, CE2 is dual home to PE3 and PE4. PE3 has a locator A3:1::/64 and a VPN SID A3:1::B100. PE4 has a locator A4:1::/64 and a VPN SID A4:1::B100. A mirror SID A4:1::3 is configured on PE4 for protecting PE3 with locator A3:1::/64.

After the mirror SID is configured on a local PE (e.g., PE4), when the local PE (e.g., BGP on the local PE) receives a prefix whose VPN SID belongs to a remote PE (e.g., PE3) with the locator that is protected by the local PE through mirror SID, the local PE (e.g., PE4) creates a mapping from the remote PE’s (e.g., PE3’s) VPN SID and the mirror SID to the local PE’s (e.g., PE4’s) VPN SID. The remote PE is protected by the local PE.

For example, local PE4 has Prefix 1.1.1.1 with VPN SID A4:1::B100, when PE4 receives prefix 1.1.1.1 with remote PE3’s VPN SID A3:1::B100, it creates a mapping from remote PE3’s VPN SID and the mirror SID (i.e., "A3:1::B100, A4:1::3") to local PE4’s VPN SID (i.e., "A4:1::B100").
Node P1’s pre-computed TI-LFA backup path for destination PE3 having locator A3:1::/64 is from P1 to PE4 having mirror SID A4:1::3. It is installed as a T.Insert transit behavior. When P1 receives a packet destined to PE3’s VPN SID A3:1::B100, in normal operations, it forwards the packet with source A1:1:: and destination PE3’s VPN SID A3:1::B100 according to the FIB using the destination PE3’s VPN SID A3:1::B100.

When PE3 fails, node P1 protects PE3 through sending the packet to PE4 via the backup path pre-computed. P1 modifies the packet before sending it to PE4. The modified packet has destination PE4 with mirror SID A4:1::3, and SRH with PE3’s VPN SID A3:1::B100 and the mirror SID A4:1::3 (i.e., "A3:1::B100, A4:1::3; SL=1").

When PE4 receives the packet, it forwards the packet to CE2 through executing END.M instruction according to the local VPN SID (i.e., A4:1::B100).

4. Extensions to IGP for Egress Protection

This section describes extensions to IS-IS and OSPF for advertising the information about SRv6 path egress protection.

4.1. Extensions to IS-IS

A new sub-TLV, called IS-IS SRv6 End.m SID sub-TLV, is defined. It is used in the SRv6 Locator TLV defined in [I-D.bashandy-isis-srv6-extensions] to advertise SRv6 Segment Identifiers (SIDs) with END.M function for SRv6 path egress protection. The SRv6 End.m SIDs inherit the topology/algorithm from the parent locator. The format of the sub-TLV is illustrated below.

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Type (TBD1)   |    Length     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Flags     |    SRv6 Endpoint Function     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         SID (16 octets)                       |
:                                                               :
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            sub-TLVs                           |
:                                                               :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

Figure 2: IS-IS SRv6 End.m SID sub-TLV
Type: TBD1 (suggested value 8) is to be assigned by IANA.

Length: variable.

 Flags: 1 octet. No flags are currently defined.

SRv6 Endpoint Function: 2 octets. Add a new endpoint function 40 for end.m SID.

SID: 16 octets. This field contains the SRv6 end.m SID to be advertised.

Two sub-TLVs are defined. One is the protected locators sub-TLV, and the other is the protected SIDs sub-TLV.

A protected locators sub-TLV is used to carry the Locators to be protected by the SRv6 mirror SID. It has the following format.

```
+---------------+---------------+---------------+
|   Type (TBD2) |   Length     |
+---------------+---------------+
| Locator-Size  | Locator (variable) |
+---------------+---------------+
```

Figure 3: IS-IS Protected Locators sub-TLV

Type: TBD2 (suggested value 1) is to be assigned by IANA.

Length: variable.

Locator-Size: 1 octet. Number of bits (1 - 128) in the Locator field.

Locator: 1-16 octets. This field encodes an SRv6 Locator to be protected by the SRv6 mirror SID. The Locator is encoded in the minimal number of octets for the given number of bits.

A protected SIDs sub-TLV is used to carry the SIDs to be protected by the SRv6 mirror SID. It has the following format.
4.2. Extensions to OSPF

Similarly, a new sub-TLV, called OSPF SRv6 End.m SID sub-TLV, is defined. It is used to advertise SRv6 Segment Identifiers (SIDs) with END.M function for SRv6 path egress protection. Its format is illustrated below.

```
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Type (TBD4)            |           Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Flags     |    SRv6 Endpoint Function     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         SID (16 octets)                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            sub-TLVs                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 5: OSPF SRv6 End.m SID sub-TLV

Type: TBD4 (suggested value 8) is to be assigned by IANA.

Length: variable.
Flags: 1 octet. No flags are currently defined.

SRv6 Endpoint Function: 2 octets. Add a new endpoint function 40 for end.m SID.

SID: 16 octets. This field contains the SRv6 end.m SID to be advertised.

Two sub-TLVs are defined. One is the protected locators sub-TLV, and the other is the protected SIDs sub-TLV.

A protected locators sub-TLV is used to carry the Locators to be protected by the SRv6 mirror SID. It has the following format.

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Type (TBD5)           |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Locator-Size  |  Locator (variable)           ˜
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Locator-Size  |  Locator (variable)           ˜
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 6: OSPF Protected Locators sub-TLV

Type: TBD5 (suggested value 1) is to be assigned by IANA.

Length: variable.

Locator-Size: 1 octet. Number of bits (1 - 128) in the Locator field.

Locator: 1-16 octets. This field encodes an SRv6 Locator to be protected by the SRv6 mirror SID. The Locator is encoded in the minimal number of octets for the given number of bits.

A protected SIDs sub-TLV is used to carry the SIDs to be protected by the SRv6 mirror SID. It has the following format.
5. Behavior for SRv6 Mirror SID

The "Endpoint with mirror protection to a vpn SID" function (End.M for short) is a variant of the End function. The End.M is used for SRv6 VPN egress protection. It is described below.

End.M: Mirror protection
When N receives a packet destined to S and S is a local End.M SID, N does:
IF NH=SRH and SL = 1 ;; Ref1
   SL--
   Map to a local VPN SID based on Mirror SID and SRH[SL] ;; Ref1
   forward according to the local VPN SID ;; Ref2
ELSE
   drop the packet

Figure 8: SRv6 Mirror SID Procedure

Ref1: An End.M SID must always be the penultimate SID.

Ref2: The rest forwarding behavior is the same as the corresponding VPN sid.

6. Security Considerations

TBD
7. IANA Considerations

TBD

8. Acknowledgements

TBD

9. References

9.1. Normative References

[I-D.bashandy-isis-srv6-extensions]

[I-D.hu-spring-segment-routing-proxy-forwarding]

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

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

[I-D.li-ospf-ospfv3-srv6-extensions]

9.2. Informative References

[I-D.bashandy-rtgwg-segment-routing-ti-lfa]
Bashandy, A., Filsfils, C., Decraene, B., Litkowski, S.,
Francois, F., daniel.voyer@bell.ca, d., Clad, F., and P.
Camarillo, "Topology Independent Fast Reroute using
Segment Routing", draft-bashandy-rtgwg-segment-routing-ti-
lfa-05 (work in progress), October 2018.

[I-D.hegde-spring-node-protection-for-sr-te-paths]
Hegde, S., Bowers, C., Litkowski, S., Xu, X., and F. Xu,
"Node Protection for SR-TE Paths", draft-hegde-spring-
node-protection-for-sr-te-paths-05 (work in progress),
July 2019.

[I-D.ietf-spring-segment-routing-policy]
Filsfils, C., Sivabalan, S., daniel.voyer@bell.ca, d.,
bogdanov@google.com, b., and P. Mattes, "Segment Routing
Policy Architecture", draft-ietf-spring-segment-routing-
policy-03 (work in progress), May 2019.

[I-D.sivabalan-pce-binding-label-sid]
Sivabalan, S., Filsfils, C., Tantsura, J., Hardwick, J.,
Previdi, S., and C. Li, "Carrying Binding Label/Segment-ID
in PCE-based Networks.", draft-sivabalan-pce-binding-
label-sid-07 (work in progress), July 2019.

[RFC5462] Andersson, L. and R. Asati, "Multiprotocol Label Switching
(MPLS) Label Stack Entry: "EXP" Field Renamed to "Traffic
Class" Field", RFC 5462, DOI 10.17487/RFC5462, February

Authors’ Addresses
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 September 11, 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
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:

- "XXXX" --> the assigned RFC value for this draft both in this draft and in the YANG models under the revision statement.
- 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:

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

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

- augment
- data model
- 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:

- 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.

- Advertises a new MAC address: Also in RFC 5227, 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.

- 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 discontinuity-time?  yang:date-and-time
      +---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@2019-02-21.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>
  Editor: Feng Zheng
    habby.zheng@huawei.com
  Editor: Bo Wu
    lana.wubo@huawei.com
  Editor: Robert Wilton
    rwilton@cisco.com
  Editor: Xiaojian Ding
    wjswal@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) 2018 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 2019-02-21 {
  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 responds to ARP requests only 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";
}

leaf interval {

type uint32 {

range "1..86400";
}

units "seconds";

description
"The interval, in seconds, between sending gratuitous ARP
packet on the interface.";
}
}

container statistics {
config false;
description
 "ARP per-interface packet statistics"

For all ARP counters, discontinuities in the value can occur at re-initialization of the management system and at other times as indicated by the value of 'discontinuity-time'."

leaf discontinuity-time {
  type yang:date-and-time;
  description
    "The time on the most recent occasion at which any one or more of this interface’s ARP counters suffered a discontinuity. If no such discontinuities have occurred since the last re-initialization of the local management subsystem, then this node contains the time the local management subsystem re-initialized itself.";
}

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 a simple example of configuring static ARP entries and dynamic learning, based on the YANG modules specified in Section 4.

5.1. Static ARP Entries

Requirement: Enable static ARP entry configuration on interface (defined in [RFC8344] ).

```xml
<config xmlns:xc="urn:ietf:params:xml:ns:netconf:base:1.0">
    <interfaces xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces">
        <ipv4 xmlns="urn:ietf:params:xml:ns:yang:ietf-ip">
            <neighbor>
                <ip-address>192.0.2.1</ip-address>
                <link-layer-address>00e0-fc01-0000</link-layer-address>
                <origin>static</origin>
            </neighbor>
        </ipv4>
    </interfaces>
</config>
```
5.2. ARP Dynamic Learning

Requirement: Disable ARP dynamic learning configuration.

```xml
<config xmlns:xc="urn:ietf:params:xml:ns:netconf:base:1.0">
  <arp xmlns="urn:ietf:params:xml:ns:yang:ietf-arp">
    <dynamic-learning>false</dynamic-learning>
  </arp>
</config>
```

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

- **interface/ipv4/arp/gratuitous-arp**: This leaf is 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.

8. Acknowledgments

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

9. References

9.1. Normative References

9.2. Informative References


Authors' Addresses

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

Email: habby.zheng@huawei.com

Zheng, et al. Expires September 11, 2019
A Simple BGP-based Mobile Routing System for the Aeronautical Telecommunications Network
draft-ietf-rtgwg-atn-bgp-02.txt

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 IPv6-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 November 15, 2019.
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/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.

The ATN/IPS is an IPv6-based overlay network configured over one or more Internetworking underlays ("INETs") maintained by aeronautical network service providers such as ARINC, SITA and Inmarsat. Each 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 any two INET partitions use the same IP protocol version nor have consistent IP addressing plans in comparison with other partitions. Instead, the ATN/IPS IPv6 overlay sees each partition as a "segment" of a link-layer topology manifested through a (virtual) bridging service known as "Spanning Partitioned Aeronautical Networks (SPAN)". Further discussion of the SPAN is found in the following sections of this document, with reference to [I-D.templin-intarea-6706bis].

The ATN/IPS further assumes that each aircraft will receive an IPv6 Mobile Network Prefix (MNP) that accompanies the aircraft wherever it travels. ICAO is further proposing to assign each aircraft an entire /56 MNP for numbering its on-board networks. ATCs and AOCs will likewise receive IPv6 prefixes, but they would typically appear in static (not mobile) deployments such as air traffic control towers, airline headquarters, etc. Throughout the rest of this document, we therefore use the term "MNP" when discussing an IPv6 prefix that is delegated to any ATN/IPS end system, including ATCs, AOCs, and aircraft. We also use the term Mobility Service Prefix (MSP) to refer to an aggregated prefix assigned to the ATN/IPS by an Internet assigned numbers authority, and from which all MNPs are delegated.
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 MNPs 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.

The s-ASBRs for each stub AS connect to a small number of c-ASBRs via dedicated high speed links and/or tunnels across the INET using industry-standard encapsulations (e.g., Generic Routing Encapsulation [GRE] [RFC2784], IPsec [RFC4301], etc.). In particular, tunneling must be used when neighboring ASBRs are separated by multiple INET hops.

The s-ASBRs engage in external BGP (eBGP) peerings with their respective c-ASBRs, and only maintain routing table entries for the MNPs currently active within the stub AS. The s-ASBRs send BGP updates for MNP 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 MNPs 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 MNPs for all partitions are known globally among all of the c-ASBRs. Each c/s-ASBR further configures a "SPAN address" which is taken from a global or unique-local IPv6 "SPAN prefix" assigned to each partition, as well as static forwarding table entries for all other prefixes in the SPAN. The SPAN addresses are used for nested encapsulation where the inner IPv6 packet is encapsulated in a SPAN header which is then encapsulated in an IP header specific to the INET partition.

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.

Spanning Partitioned Aeronautical Networks (SPAN)
A virtual layer 2 bridging service that presents a unified link view to the ATN/IPS overlay even though the underlay may consist of multiple INET partitions. The SPAN is manifested through nested encapsulation in which IPv6 packets from the ATN/IPS are first encapsulated in SPAN headers which are then encapsulated in INET headers. In this way, packets sent from a source can be conveyed over the SPAN even though there may be many underlying INET partitions in the path to the destination.

SPAN Autonomous System
A "hub-of-hubs" autonomous system maintained through peerings between the core autonomous systems of different SPAN 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
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
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
An ANET/INET border node that acts as a transparent intermediary between Clients and s-ASBRs. From the Client’s perspective, the Proxy presents the appearance that the Client is communicating directly with the s-ASBR. From the s-ASBR’s perspective, the Proxy 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 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 overlay does not interact with the underlying INET BGP routing systems, and only a small and unchanging set of MSPs are advertised externally instead of the full dynamically changing set of MNPs.

Within each INET partition, one or more s-ASBRs connect each stub AS to the INET partition core using a shared 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. Globally-unique public ASNs could be assigned, e.g., either according to the standard 16-bit ASN format or the 32-bit ASN scheme defined in [RFC6793].

The c-ASBRs use iBGP to maintain a synchronized consistent view of all active MNPs currently in service within the INET partition. Figure 1 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.
In the reference deployment, each s-ASBR maintains routes for active MNPs that currently belong to its stub AS. In response to "Inter-domain" mobility events, each s-ASBR will dynamically announces new MNPs and withdraws departed MNPs 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 will maintain forwarding table
entries only for the MNPs that are currently active, and packets destined to all other MNPs will correctly incur ICMPv6 Destination Unreachable messages [RFC4443] due to the black hole route. (This is the same behavior as for ordinary BGP routers in the Internet when they receive packets for which there is no route available.) The c-ASBRs do not send eBGP updates for MNPs 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 MNPs currently within their stub ASes) and they forward all other packets to c-ASBRs which have full topology knowledge.

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" SPAN AS. The SPAN AS contains the global knowledge of all MNPs deployed worldwide, and supports ATN/IPS overlay communications between nodes located in different INET partitions by virtue of SPAN encapsulation. Figure 2 shows a reference SPAN topology.

```
(...)
```

Figure 2: The SPAN
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 MNPs 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 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 at the ANET/INET boundary.

Figure 3 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 at the edge of the ANET.

When a Client logs into an ANET it specifies a nearby s-ASBR that it has selected to connect to the ATN/IPS. (Selection of a nearby s-ASBR could be through consulting a geographically-keyed static host file, through a DNS lookup, through a network query response, etc.) The login process is transparently brokered by a Proxy at the border of the ANET, which then conveys the connection request to the s-ASBR via tunneling across the SPAN. The s-ASBR then registers the address of the Proxy as the address for the Client, and the Proxy forwards the s-ASBR’s reply to the Client. If the Client connects to multiple
ANETs, the s-ASBR will register the addresses of all Proxies as addresses through which the Client can be reached.

The s-ASBR represents all of its active Clients as MNP routes in the ATN/IPS BGP routing system. The s-ASBR’s stub AS therefore consists of the set of all of its active Clients (i.e., the stub AS is a logical construct and not a physical construct). The s-ASBR injects the MNPs of its active Clients and withdraws the MNPs of its departed Clients via BGP updates to c-ASBRs, which further propagate the MNPs to other c-ASBRs within the SPAN AS. Since Clients are expected to remain associated with their current s-ASBR for extended periods, the level of MNP 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 tunnel segments in the forwarding path. In many cases, it would be desirable to eliminate extraneous tunnel segments from this "dogleg" route so that packets can traverse a minimum number of tunneling hops across the SPAN. ATN/IPS end systems could therefore employ a route optimization service according to the mobility service employed (see: Section 7).

A route optimization example is shown in Figure 4 and Figure 5 below. In the first figure, multiple tunneled segments between Proxys 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 Proxys without involving the ASBRs.
Figure 4: Dogleg Route Before Optimization
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 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 (800 bits), 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 and should 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.

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 and operational standpoint, the implementation should provide isolation between the two BGP routing domains (e.g., separate BGP instances).

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] and AERO [I-D.templin-intarea-6706bis] 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 MNPs 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], TLS [RFC5246], 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.

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

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 Information Technology (BIT) MobileNet program.

The following individuals contributed insights that have improved the document: Erik Kline, Hubert Kuenig, Tony Li, Alexandre Petrescu, Pascal Thubert, Tony Whyman.
12. References

12.1. Normative References


12.2. Informative References


Appendix A. BGP Convergence Considerations

Experimental evidence has shown that BGP convergence time required for when an MNP 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 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 a SPAN address for the new s-ASBR. The SPAN 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 >>

Changes from -01 to -02:

- introduced the SPAN and the concept of Internetwork partitioning
- new terms "ANET" (for (Radio) Access Network) and "INET" (for Internetworking underlay)
- new appendix on BGP convergence considerations

Changes from -00 to -01:

- incorporated clarifications due to list comments and questions.
- new section 7 on Stub AS Mobile Routing Services
- updated references, and included new reference for MIPv6 and LISP

Status as of 08/30/2018:

- 'draft-templin-atn-bgp' becomes 'draft-ietf-rtgwg-atn-bgp'

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
Gap Analysis of Dynamic Networks to Hybrid Cloud DCs

draft-ietf-rtgwg-net2cloud-gap-analysis-02

Abstract

This document analyzes the technological gaps when using SD-WAN to
dynamically interconnect workloads and applications hosted in
various third-party cloud data centers.

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/1id-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 December 19, 2019.
1. Introduction

[Net2Cloud-Problem] describes the problems that enterprises face today in transitioning their IT infrastructure to support digital
economy, such as connecting enterprises’ branch offices to dynamic workloads in different Cloud DCs.

This document analyzes the technological gaps to interconnect dynamic workloads & apps hosted in cloud data centers that the enterprise’s VPN service provider may not own/operate or may be unable to provide the enterprise with the required connectivity to access such locations. When VPN service providers have insufficient bandwidth to reach a location, SD-WAN techniques can be used to aggregate bandwidth of multiple networks, such as MPLS VPNs or the Public Internet to achieve better performance. This document primarily focuses on the technological gaps raised by using SD-WAN techniques to connect enterprise premises to cloud data centers operated by third parties.

For the sake of readability, a SD-WAN edge, a SD-WAN endpoint, C-PE, or CPE are used interchangeably throughout this document. However, each term has some minor emphasis, especially when used in other related documents:

. SD-WAN Edge: could include multiple devices (virtual or physical);
. SD-WAN endpoint: to refer to a WAN port of SD-WAN devices or a single SD-WAN device;
. C-PE: more for provider owned SD-WAN edge, e.g. for SECURE-EVPN’s PE based VPN, when PE is the edge node of SD-WAN;
. CPE: more for enterprise owned SD-WAN edge.

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 SD-WAN controller to manage SD-WAN 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

SD-WAN: Software Defined Wide Area Network, "SD-WAN" refers to the solutions of pooling WAN bandwidth from multiple underlay networks to get better WAN bandwidth management, visibility & control. When the underlay is a private network, traffic may be forwarded without any additional encryption; when the underlay networks are public, such as the Internet, some traffic needs to be encrypted when passing through (depending on user-provided policies).

3. Gap Analysis of C-PEs WAN Port Registration

SD-WAN technology has emerged as means to dynamically and securely interconnect the OnPrem branches with the workloads instantiated in Cloud DCs that do not have direct connectivity to BGP/MPLS VPN PEs or have very limited bandwidth.

Some SD-WAN networks use the NHRP protocol [RFC2332] to register WAN ports of SD-WAN edges with a "Controller" (or NHRP server), which then has the ability to map a private VPN address to a public IP address of the destination node. DSVPN [DSVPN] or DMVPN [DMVPN] are used to establish tunnels between WAN ports of SD-WAN edge nodes.

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

- Interworking with the MPLS VPN control plane. A SD-WAN 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 before being sent.
- Scalability: NHRP/DSVPN/DMVPN works 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 C-PE’s owner to be informed of how/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 addresses, such as the NAT type, Private address, Public address, Private port, Public port, etc.

[BGP-SDWAN-PORT] describes how SD-WAN edge nodes use BGP to register their WAN ports properties to the SD-WAN controller, which then propagates the information to other SD-WAN edge nodes that are authenticated and authorized to communicate with them.

4. Aggregating VPN paths and Internet paths

Most likely, enterprises (especially the largest ones) already have their CPEs interconnected by providers’ VPNs, based upon VPN techniques such as EVVPN, L2VPN, or L3VPN. The VPN can be PE-based or CPE-based. The commonly used PE-based VPNs have CPE directly attached to PEs, therefore the communication between CPEs and PEs is considered as secure. MP-BGP is used to learn & distribute routes among CPEs, even though sometimes routes among CPEs are statically configured on the CPEs.

To aggregate paths over the Internet and paths over the VPN, the C-PEs need to have some WAN ports connected to the PEs of the VPNs and other WAN ports connected to the Internet. It is necessary for the CPEs to use a protocol so that they can register the WAN port properties with their SD-WAN Controller(s): this information conditions the establishment and the maintenance of IPsec SA associations among relevant C-PEs.
When using NHRP for registration purposes, C-PEs need to run two separate control planes: EVPN&BGP for CPE-based VPNs, and NHRP & DSVPN/DMVPN for ports connected to the Internet. Two separate control planes not only add complexity to C-PEs, but also increase operational cost.

```
+---+  +---------+  packets encrypted over  +---+  +----+
| TN3|--|         A1-----+ Untrusted    +------ B1     |--| TN1|
+++++| C-PE |A2-\      +----| C-PE  +----+
+++++| A A3--++  +---------B2  B +----+
| TN2|--|         | PE+--------------+PE |---B3     |--| TN3|
+++++| +---------+  trusted    +---+   ++---+++  +----+
|     |         |            | W  |            |     |
|     |         |            |     |            |     |
|     |         |            |     |            |     |
|     |         |            |     |            |     |
|     |         |            |     |            |     |
|     |         |            |     |            |     |
|     |         |            |     |            |     |
|     |         |            |     |            |     |
+----+  +---------+  packets    +---+  +----+
| TN1|--|         C1--|PE| go natively |PE|-- D1     |--| TN1|
+++++| C-PE |C2------| without encry+---| C-PE  +----+
|     | |     |            |     |            |     |
|     |     |            |     |            |     |
|     |     |            |     |            |     |
|     |     |            |     |            |     |
|     |     |            |     |            |     |
|     |     |            |     |            |     |
|     |     |            |     |            |     |
|     |     |            |     |            |     |
|     |     |            |     |            |     |
|     |     |            |     |            |     |
|     |     |            |     |            |     |
|     |     |            |     |            |     |
|     |     |            |     |            |     |
+----+  +---------+  without encrypt over  +----+
| TN2|--|         C3--| Untrusted  +----+ D2     |--| TN2|
+++++| +---------+  ++---+++  +----+
|     |         |            |     |
|     |         |            |     |
|     |         |            |     |
|     |         |            |     |
|     |         |            |     |
+----+  +---------+

Figure 1: CPEs interconnected by VPN paths and Internet Paths

4.1. Key Control Plane Components of SD-WAN

As described in [BGP-SDWAN-Usage], the SD-WAN Overlay Control Plane has three distinct properties:

- SD-WAN node’s WAN Port Property registration to the SD-WAN Controller.
  - To inform the SD-WAN controller and authorized peers of the WAN port properties of the C-PE [SDWAN-Port]. When the WAN ports are assigned private addresses, this step can register the type of NAT that translates private addresses into public ones.

- Controller facilitated IPsec SA management and NAT information distribution
- Establishing and Managing the topology and reachability for services attached to the client ports of SD-WAN nodes.
  - This is for the overlay layer’s route distribution, so that a C-PE can populate its overlay routing table with entries that identify the next hop for reaching a specific route/service attached to remote nodes. [SECURE-EVPN] describes EVPN and other options.

4.2. Using BGP Tunnel-Encap

RFC5512 and [Tunnel-Encap] describe methods to construct BGP UPDATE messages that advertise endpoints’ tunnel encapsulation capability and the respective attached client routes, so that the peers that receive of the BGP UPDATE can establish appropriate tunnels with the endpoints for the aforementioned routes. RFC5512 uses the Endpoint Address subTLV, whereas [Tunnel-Encap] uses Remote Endpoint Address subTLV to indicates address of the tunnel endpoint which can be an IPv4 or an IPv6 address. There are Tunnel Encapsulation attribute subTLVs to indicate the supported encapsulation types, such as L2TPv3, GRE, VxLAN, IP-in-IP, etc.

[Tunnel-Encap] removed SAFI =7 (which was specified by RFC5512) for distributing encapsulation tunnel information. [Tunnel-Encap] requires that tunnels need to be associated with routes.

There is also the Color sub-TLV to describe customer-specified information about the tunnels (which can be creatively used for SD-WAN).

Here are some of the gaps using [Tunnel-Encap] to control SD-WAN Tunnels:

- [Tunnel-Encap] doesn’t have the functionality that would help the C-PE to register its WAN Port properties.
- A SD-WAN tunnel, e.g. IPsec-based, requires a negotiation between the tunnel’s end points for supported encryption algorithms and tunnel types before it can be properly established, whereas [Tunnel-Encap] only allow the announcement of one endpoint’s supported encapsulation capabilities for specific attached routes.
and no negotiation between tunnel end points is needed. The establishment of a SD-WAN tunnel can fail, e.g., in case the two endpoints support different encryption algorithms. That is why a SD-WAN tunnel needs to be established and maintained independently from advertising client routes attached to the edge node.

- [Tunnel-Encap] requires all tunnels updates are associated with routes. There can be many client routes associated with the SD-WAN IPsec tunnel between two C-PEs’ WAN ports; the corresponding destination prefixes (as announced by the aforementioned routes) may also be reached through the VPN underlay without any encryption. A more realistic approach to separate SD-WAN tunnel management from client routes association with the SD-WAN tunnels.

- When SD-WAN tunnel and clients routes are separate, the SD-WAN Tunnel establishment may not have routes associated. There is a suggestion on using a "Fake Route" for a SD-WAN node to use [Tunnel-Encap] to advertise its SD-WAN tunnel end-points properties. However, using "Fake Route" can raise some design complexity for large SD-WAN networks with many tunnels. For example, for a SD-WAN network with hundreds of nodes, with each node having many ports & many endpoints to establish SD-WAN tunnels with their corresponding peers, the node would need as many "fake addresses". For large SD-WAN networks (such as those comprised of more than 10000 nodes), each node might need 10’s thousands of "fake addresses", which is very difficult to manage and requires lots of configuration tasks to get the nodes properly set up.

- [Tunnel-Encap] does not have any field to carry detailed information about the remote C-PE, such as Site-ID, System-ID, Port-ID

- [Tunnel-Encap] Does not have any field to carry IPsec attributes for the SD-WAN edge nodes to establish IPsec Security Associations with others. It does not have any proper way for two peer CPEs to negotiate IPsec keys either, based on the configuration sent by the Controller.

- [Tunnel-Encap] does not have any field to indicate the UDP NAT private address <-> public address mapping

- C-PEs tend to communicate with a subset of the other C-PEs, not all the C-PEs need to be connected through a mesh topology. Without any BGP extension, many nodes can get dumped with too much
information coming from other nodes that they never need to communicate with.

4.3. SECURE-L3VPN/EVPN

[SECURE-L3VPN] describes how to extend the 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-protected packets are forwarded to the Internet or to other backbone network thereby eliminating the need for MPLS transport in the backbone.

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

[SECURE-EVPN] describes a solution where point-to-multipoint BGP signaling is used in the control plane for SDWAN Scenario #1. 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 without IKEv2 point-to-point signaling or any other direct peer-to-peer session establishment messages.

Both [SECURE-L3VPN] and [SECURE-EVPN] are useful, however, they both miss the aspects of aggregating VPN and Internet underlays. In summary:

- These documents do not address the scenario of C-PE having some ports facing VPN PEs and other ports facing the Internet.
- 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. In SD-WAN, Zero Touch Provisioning is expected. Manual configuration is not an option, as it contradicts the objectives of SD-WAN to automate configuration tasks.
- For RR communication with C-PEs, this draft only mentions IPsec. Missing TLS/DTLS.
- The draft assumes that C-PEs and RR are connected with an IPsec tunnel. With zero touch provisioning, we need an automatic way to synchronize the IPsec SAs between C-PEs and RR. The draft assumes:
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 draft 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. Let us assume that the RR is responsible for creating the key for all endpoints: When one endpoint is compromised, all other connections will be impacted.

4.4. 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 CPEs to support means to determine whether traffic sent by remote peers is legitimate to prevent spoofing attacks.

5. CPEs not directly connected to VPN PEs

Because of the ephemeral property of the selected Cloud DCs, an enterprise or its network service provider may not have direct connections to the Cloud DCs that are used for hosting the enterprise’s specific workloads/Apps. Under those circumstances, SD-WAN is a very flexible choice to interconnect the enterprise on-premises data centers & branch offices to its desired Cloud DCs.

However, SD-WAN paths established over the public Internet can have unpredictable performance, especially over long distances and across operators’ domains. Therefore, it is highly desirable to steer as much as possible the portion of SD-WAN paths over service provider VPN (e.g., enterprise’s existing VPN) that have guaranteed SLA to minimize the distance or the number of segments over the public Internet.
MEF Cloud Service Architecture [MEF-Cloud] also describes a use case of network operators that uses SD-WAN over LTE or the public Internet for last mile access where the VPN service providers cannot necessarily provide the required physical infrastructure.

Under those scenarios, one or two of the SD-WAN endpoints may not be directly attached to the PEs of a VPN Domain.

When using SD-WAN to connect the enterprise’s existing sites with the workloads hosted in Cloud DCs, the corresponding CPEs have to be upgraded to support SD-WAN. 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 integrates SD-WAN with traditional MPLS-based VPNs, to extend 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 MPLS VPN can be designated to connect to the remote workloads via SD-WAN 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 existing CPEs. The only CPE that needs to support SD-WAN would be a virtualized CPE instantiated within the cloud DC.
In Figure 3, the optimal Cloud DC to host the workloads (as a function of the proximity, capacity, pricing, or other criteria chosen by the enterprises) does not have a direct connection to the PEs of the MPLS VPN that interconnects the enterprise’s existing sites.
5.1. Floating PEs to connect to Remote CPEs

To extend MPLS VPNs to remote CPEs, it is necessary to establish secure tunnels (such as IPsec tunnels) between the Floating PEs and the remote CPEs.

Even though a set of PEs can be manually selected to act as the floating PEs for a specific cloud data center, there are no standard protocols for those PEs to interact with the remote CPEs (most likely virtualized) instantiated in the third party cloud data centers (such as exchanging performance or route information).

When there is more than one fPE available for use (as there should be for resiliency purposes or the ability to support multiple cloud DCs geographically scattered), it is not straightforward to designate an egress fPE to remote CPEs based on applications. There is too much applications’ traffic traversing PEs, and it is not feasible for PEs to recognize applications from the payload of packets.

5.2. 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.

A SD-WAN edge node can solicit a STUN (Session Traversal of UDP Through Network Address Translation RFC 3489) Server to get the NAT property, the public IP address and the Public Port number so that such information can be communicated to the relevant peers.

5.3. Complexity of using 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 complications in extending BGP from MPLS VPN PEs to remote CPEs via any access path (e.g., Internet).

The path between the remote CPEs and VPN PEs that maintain VPN routes may very well traverse untrusted nodes.
EBGP Multi-hop design requires static configuration on both peers. 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.

5.4. Designated Forwarder to the remote edges

Among the multiple floating PEs that are reachable from a remote CPE, 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.

Therefore, it is necessary to designate one floating PE as the CPE’s Designated Forwarder, similar to TRILL’s Appointed Forwarders [RFC6325].
MPLS VPNs do not have features like TRILL’s Appointed Forwarders.

5.5. Traffic Path Management

When there are multiple floating PEs that have established IPsec tunnels with the remote CPE, the remote CPE can forward outbound traffic to the Designated Forwarder PE, which in turn forwards traffic to egress PEs and then to the final destinations. However, it is not straightforward for the egress PE to send back the return traffic to the Designated Forwarder PE.

Example of Return Path management using Figure 3 above.

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

6. Manageability Considerations

Zero touch provisioning of SD-WAN edge nodes should be a major feature of SD-WAN deployments. It is necessary for a newly powered up SD-WAN edge node to establish a secure connection (by means of TLS, DTLS, etc.) with its controller.

7. Security Considerations

The intention of this draft is to identify the gaps in current and proposed SD-WAN approaches that can address requirements identified in [Net2Cloud-problem].

Several of these approaches have gaps in meeting enterprise security requirements when tunneling their traffic over the Internet, since this is the purpose of SD-WAN. See the individual sections above for further discussion of these security gaps.
8. IANA Considerations

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

9. References

9.1. Normative References


9.2. Informative References


[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:

[DSVPN] Dynamic Smart VPN:


10. Acknowledgments

Acknowledgements to xxx for his review and contributions.

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

Linda Dunbar
Futurewei
Email: ldunbar@futurewei.com

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

Christian Jacquenet
Orange
Rennes, 35000
France
Email: Christian.jacquenet@orange.com
Dynamic Networks to Hybrid Cloud DCs Problem Statement  
draft-ietf-rtgwg-net2cloud-problem-statement-04

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).

It examines some of the approaches interconnecting cloud DCs with enterprises’ on-premises DCs & branch offices. This document also describes some of 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.
1. Introduction
   1.1. On the evolution of Cloud DC connectivity
   1.2. The role of SD-WAN techniques in Cloud DC connectivity

2. Definition of terms

3. Interconnecting Enterprise Sites with Cloud DCs
   3.1. Multiple connections to workloads in a Cloud DC
   3.2. Interconnect Private and Public Cloud DCs
   3.3. Desired Properties for Networks that interconnect Hybrid Clouds

4. Multiple Clouds Interconnection
   4.1. Multi-Cloud Interconnection
   4.2. Desired Properties for Multi-Cloud Interconnection
1. Introduction

1.1. On the evolution of Cloud DC connectivity

The ever-increasing use of cloud applications for communication services change the way corporate business works and shares information. Such cloud applications use resources hosted in third party DCs that also host services for other customers.

With the advent of widely available third-party cloud DCs in diverse geographic locations and the advancement of tools for monitoring and predicting application behaviors, it is technically feasible for enterprises to instantiate applications and workloads in locations that are geographically closest to their end-users. Such proximity improves 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, an enterprise may wish to take advantage of more and more business applications offered by third party private cloud DCs.

Most of those enterprise branch offices & on-premises data centers are already connected via VPNs, such as MPLS-based L2VPNs and L3VPNs. Then connecting to the cloud-hosted resources may not be straightforward if the provider of the VPN service does not have direct connections to the corresponding cloud DCs. Under those circumstances, the enterprise can upgrade the CPEs deployed in its...
various premises to utilize SD-WAN techniques to reach cloud resources (without any assistance from the VPN service provider), or wait for their VPN service provider to make new agreements with data center providers to connect to the cloud resources. Either way has additional infrastructure and operational costs.

In addition, more enterprises are moving towards hybrid cloud DCs, i.e. owned or operated by different Cloud operators, to maximize the benefits of geographical proximity, elasticity and special features offered by different cloud DCs.

1.2. The role of SD-WAN techniques in Cloud DC connectivity

This document discusses the issues associated with connecting enterprise’s workloads/applications instantiated in multiple third-party data centers (a.k.a. Cloud DCs) and its on-prem data centers. Very often, the actual Cloud DCs that host the workloads/applications can be transient.

SD-WAN, initially launched to maximize bandwidths between locations by aggregating multiple paths managed by different service providers, has expanded to include flexible, on-demand, application-based connections established over any networks to access dynamic workloads in Cloud DCs.

Therefore, this document discusses the use of SD-WAN techniques to improve enterprise-to-cloud DC and cloud DC-to-cloud DC connectivity.

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).

SD-WAN: Software Defined Wide Area Network. In this document, "SD-WAN" refers to the solutions of pooling WAN bandwidth from multiple underlay networks to get better WAN bandwidth management, visibility & control. When the underlay networks are private networks, traffic can traverse without additional encryption; when the underlay networks are public, such as Internet, some traffic needs to be encrypted when traversing through (depending on user provided policies).

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. Interconnecting Enterprise Sites with Cloud DCs

3.1. Multiple connections to workloads in a Cloud DC

Most Cloud operators offer some type of network gateway through which an enterprise can reach their workloads hosted in the Cloud DCs. For example, 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.

As an example, some branch offices of an enterprise can connect to over the Internet to reach AWS’s vGW via IPsec tunnels. Other branch offices of the same enterprise can connect to AWS DirectConnect via a private network (without any encryption). It is important for enterprises to be able to observe the specific behaviors when connected by different connections.

Figure below shows an example of some tenants’ 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. vR1 uses IPsec to establish secure tunnels over the Internet to avoid paying extra fees for the IPsec features provided by AWS vGW. Some tenants can deploy separate virtual routers to connect to internet traffic and to traffic from the secure channels from vGW and DirectConnect, e.g. vR1 & vR2. Others may have one virtual router connecting to both types of traffic. Customer Gateway can be customer owned router or ports physically connected to AWS Direct Connect GW.
3.2. Interconnect Private and Public Cloud DCs

It is likely that hybrid designs will become the rule for cloud services, as more enterprises see the benefits of integrating public and private cloud infrastructures. However, enabling the growth of hybrid cloud deployments in the enterprise requires fast and safe interconnection between public and private cloud services. For an enterprise to connect to applications & workloads hosted in multiple Cloud DCs, the enterprise can use IPsec tunnels established over the Internet or a (virtualized) leased line service to connect its on-premises gateways to each of the Cloud DC’s gateways, virtual...
routers instantiated in the Cloud DCs, or any other suitable design (including a combination thereof).

Some enterprises prefer to instantiate their own virtual CPEs/routers inside the Cloud DC to connect the workloads within the Cloud DC. Then an overlay path is established between customer gateways to the virtual CPEs/routers for reaching the workloads inside the cloud DC.

3.3. Desired Properties for Networks that interconnect Hybrid Clouds

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

- High availability to access all workloads in the desired cloud DCs.
  Many enterprises include cloud infrastructures in their disaster recovery strategy, e.g., by enforcing periodic backup policies within the cloud, or by running backup applications in the Cloud, etc. Therefore, the connection to the cloud DCs may not be permanent, but rather needs to be on-demand.

- 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.
  Some enterprises have front-end web portals running in cloud DCs and database servers in their on-premises DCs. Those front-end web portals need to be reachable from the public Internet. The backend connection to the sensitive data in database servers hosted in the on-premises DCs might need secure connections.

- Scalable security management. IPsec is commonly used to interconnect cloud gateways with CPEs deployed in the enterprise premises. For enterprises with a large number or branch offices, managing the IPsec’s Security Associations among many nodes can be very difficult.
4. Multiple Clouds Interconnection

4.1. Multi-Cloud Interconnection

Enterprises today can instantiate their workloads or applications in Cloud DCs owned by different Cloud providers, e.g. AWS, Azure, GoogleCloud, Oracle, etc. Interconnecting those workloads involves three parties: The Enterprise, its network service providers, and the Cloud providers.

All Cloud Operators offer secure ways to connect enterprises’ on-prem sites/DCs with their Cloud DCs.

Some Cloud Operators allow enterprises to connect via private networks. For example, AWS’s DirectConnect allows enterprises to use third-party provided private Layer 2 path from enterprises’ GW to AWS DirectConnect GW. 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.

All those connectivity options are between Cloud providers’ DCs and the Enterprises, but not between cloud DCs. For example, to connect applications in AWS Cloud to applications in Azure Cloud, there must be 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 strong swan 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 VNDC, 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. [BGP-SDWAN] describes one method. Other methods are worth exploring.

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.

Other protocols such as BGP can be used, as described in [BGP-SDWAN].

4.2. Desired Properties for Multi-Cloud Interconnection

Different Cloud Operators have different APIs to access their Cloud resources. 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. SDWAN is positioned to become that network fabric enabling Cloud DCs to be dynamically attached or detached. But the reality is that different Cloud Operators have different access methods, and Cloud DCs might be geographically far apart. More Cloud connectivity problems are described in the subsequent sections.

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.

5. 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, since 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 usually 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.

- Extensive usage of Overlay by Cloud DCs:
Many cloud DCs use an overlay to connect their gateways to the workloads located inside the DC. There is currently no standard that specifies the interworking between the Cloud Overlay and the enterprise’ existing underlay networks. One of the characteristics of overlay networks is that some of the WAN ports of the edge nodes connect to third party networks. There is therefore a need to propagate WAN port information to remote authorized peers in third party network domains in addition to route propagation. Such an exchange cannot happen before communication between peers is properly secured.

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).

6. 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. 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 usually have multiple sites and multiple data centers. For workloads and apps hosted in cloud DCs, multiple sites need to communicate securely with those cloud workloads and apps. This section documents some of the issues associated with using IPsec tunnels to connect enterprise premises with cloud gateways.

6.1. Complexity of multi-point any-to-any interconnection

The dynamic workload instantiated in cloud DC needs to communicate with multiple branch offices and on-premises data centers. Most
enterprises need multi-point interconnection among multiple locations, which can be provided by means of MPLS L2/L3 VPNs.

Using IPsec overlay paths to connect all branches & on-premises data centers to cloud DCs requires CPEs to manage routing among Cloud DCs gateways and the CPEs located at other branch locations, which can dramatically increase the complexity of the design, possibly at the cost of jeopardizing the CPE performance.

The complexity of requiring CPEs to maintain routing among other CPEs is one of the reasons why enterprises migrated from Frame Relay based services to MPLS-based VPN services.

MPLS-based VPNs have their PEs directly connected to the CPEs. Therefore, CPEs only need to forward all traffic to the directly attached PEs, which are therefore responsible for enforcing the routing policy within the corresponding VPNs. Even for multi-homed CPEs, the CPEs only need to forward traffic among the directly connected PEs. However, when using IPsec tunnels between CPEs and Cloud DCs, the CPEs need to compute, select, establish and maintain routes for traffic to be forwarded to Cloud DCs, to remote CPEs via VPN, or directly.

6.2. 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.

6.3. Scaling Issues with IPsec Tunnels

IPsec can achieve secure overlay connections between two locations over any underlay network, e.g., between CPEs and Cloud DC Gateways.
If there is only one enterprise location connected to the cloud gateway, a small number of IPsec tunnels can be configured on-demand between the on-premises DC and the Cloud DC, which is an easy and flexible solution.

However, for multiple enterprise locations to reach workloads hosted in cloud DCs, the cloud DC gateway needs to maintain multiple IPsec tunnels to all those locations (e.g., as a hub & spoke topology). 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 not only very expensive (because Cloud Operators usually charge their customers based on connections), but can be very processing intensive for the gateway. 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 here is key distribution and maintenance of a key server, etc.

7. Problems of Using SD-WAN to connect to Cloud DCs

SD-WAN can establish parallel paths over multiple underlay networks between two locations on-demand, for example, to support the connections established between two CPEs interconnected by a traditional MPLS VPN ([RFC4364] or [RFC4664]) or by IPsec [RFC6071] tunnels.

SD-WAN lets enterprises augment their current VPN network with cost-effective, readily available Broadband Internet connectivity, enabling some traffic offloading to paths over the Internet according to differentiated, possibly application-based traffic forwarding policies, or when the MPLS VPN connection between the two locations is congested, or otherwise undesirable or unavailable.

7.1. SD-WAN among branch offices vs. interconnect to Cloud DCs

SD-WAN interconnection of branch offices is not as simple as it appears. For an enterprise with multiple sites, using SD-WAN overlay paths among sites requires each CPE to manage all the addresses that local hosts have the potential to reach, i.e., map internal VPN addresses to appropriate SD-WAN paths. This is similar to the complexity of Frame Relay based VPNs, where each CPE needed to maintain mesh routing for all destinations if they were to avoid an extra hop through a hub router. Even though SD-WAN CPEs can get assistance from a central controller (instead of running a routing...
protocol) to resolve the mapping between destinations and SD-WAN paths, SD-WAN CPEs are still responsible for routing table maintenance as remote destinations change their attachments, e.g., the dynamic workload in other DCs are de-commissioned or added.

Even though originally envisioned for interconnecting branch offices, SD-WAN offers a very attractive way for enterprises to connect to Cloud DCs.

The SD-WAN for interconnecting branch offices and the SD-WAN for interconnecting to Cloud DCs have some differences:

- SD-WAN for interconnecting branch offices usually have two end-points (e.g., CPEs) controlled by one entity (e.g., a controller or management system operated by the enterprise).
- SD-WAN for Cloud DC interconnects may consider CPEs owned or managed by the enterprise, while remote end-points are being managed or controlled by Cloud DCs (For the ease of description, let’s call such CPEs asymmetrically-managed CPEs).
Cloud DCs may have different entry points (or devices) with one entry point that terminates a private direct connection (based upon a leased line for example) and other entry points being devices terminating the IPsec tunnels, as shown in Figure 2.

Therefore, the SD-WAN design becomes asymmetric.

Figure 2: Different Underlays to Reach Cloud DC

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

In order 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


13. Acknowledgments

Many thanks to Alia Atlas, Chris Bowers, 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
Independent
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
Abstract

This document defines a YANG data model 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.

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
1. Introduction

This document describes a YANG [RFC6020] [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, in order to allow operators to manage policy configuration in a consistent, intuitive way in heterogeneous 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 a number of 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 an intuitive and straightforward way to express policy without sacrificing flexibility. A side affect of this design decision is that legacy methods for expressing policies are not considered. Such methods could be added as an augmentation to the model if needed.

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.

The following terms are defined in [RFC8342]:

- client
- server
- configuration
- system state
- operational state
The following terms are defined in [RFC7950]:

- action
- augment
- container
- container with presence
- data model
- data node
- feature
- leaf
- list
- mandatory node
- module
- schema tree
- 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.
3. Model overview

The routing policy module has three main parts:

- A generic framework to express policies as sets of related conditions and actions. This includes match sets and actions that are useful across many routing protocols.

- A structure that allows routing protocol models to add protocol-specific policy conditions and actions through YANG augmentations. 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].

- A reusable grouping 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.

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 effect multiple changes to route attributes, or indicate a final disposition of accepting or rejecting the route. This structure is shown below.
4.1. Defined sets for policy matching

The models provide a set 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 supported defined sets include:

- prefix sets - define a set of IP prefixes, each with an associated CIDR netmask range (or exact length)

- neighbor sets - define a set of neighboring nodes by their IP addresses. These sets are used for selecting routes based on the neighbors advertising the routes.

- tag set - define a set of generic tag values that can be used in matches for filtering routes

The model structure for defined sets is shown below.
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.

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

- **ALL** - match is true only if the given value matches all members of the set.
- **ANY** - match is true if the given value matches any member of the set.
- **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 also 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 install-protocol-eq?
          +--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?
```

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.
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 chain is reached, the default route disposition action is returned (i.e., boolean false for reject-route unless an alternate default action is specified for the chain). Consequently, a subroutine cannot explicitly accept or reject a route. Rather it merely provides an indication that ‘call-policy’ condition returns boolean true or false indicating whether or not the condition matches. 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, some major 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) should be discouraged.
5. Policy evaluation

Evaluation of each policy definition proceeds by evaluating its corresponding individual policy statements in order. 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 definitions in the chain are evaluated.

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).

Note that the route’s pre-policy attributes are always used for testing policy statement conditions. In other words, if actions modify the policy application specific attributes, those modifications are not used for policy statement conditions.

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) that have an associated direction (import or export) with respect to the routing context in which they are defined. 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. 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.

An example of this is shown below, in which the BGP configuration model in [I-D.ietf-idr-bgp-model] adds new defined sets to match on community values or AS paths. The model similarly augments BGP-specific conditions and actions in the corresponding sections of the routing policy model.

```plaintext
+++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 masklength-lower
                               masklength-upper]
            +++rw ip-prefix inet:ip-prefix
            +++rw masklength-lower uint8
            +++rw masklength-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 bgp-pol:bgp-defined-sets
      +++rw bgp-pol:community-sets
        +++rw bgp-pol:community-set* [community-set-name]
          +++rw bgp-pol:community-set-name    string
          +++rw bgp-pol:community-member*    union
      +++rw bgp-pol:ext-community-sets
        +++rw bgp-pol:ext-community-set* [ext-community-set-name]
          +++rw bgp-pol:ext-community-set-name    string
          +++rw bgp-pol:ext-community-member*    union
      +++rw bgp-pol:as-path-sets
        +++rw bgp-pol:as-path-set* [as-path-set-name]
          +++rw bgp-pol:as-path-set-name    string
          +++rw bgp-pol:as-path-set-member*    string
  +++rw policy-definitions
```
8. Security Considerations

Routing policy configuration has a significant impact on network operations, and, as such, any related model carries potential security risks.

YANG data models are generally designed to be used with the NETCONF protocol over an SSH transport. This provides an authenticated and secure channel over which to transfer configuration and operational data. Note that use of alternate transport or data encoding (e.g., JSON over HTTPS) would require similar mechanisms for authenticating and securing access to configuration data.

Most of the data elements in the policy model could be considered sensitive from a security standpoint. Unauthorized access or invalid data could cause major disruption.

9. IANA Considerations

This YANG data model and the component modules currently use a temporary ad-hoc namespace. If and when it is placed on redirected for the standards track, an appropriate namespace URI will be registered in the IETF XML Registry" [RFC3688]. The routing policy YANG modules will be registered in the "YANG Module Names" registry [RFC6020].
10. YANG modules

The routing policy model is described by the YANG modules in the sections below.

10.1. Routing policy model

<CODE BEGINS> file "ietf-routing-policy@2019-03-06.yang"
module ietf-routing-policy {
  yang-version "1.1";
  prefix rt-pol;

  import ietf-inet-types {
    prefix "inet";
  }

  import ietf-yang-types {
    prefix "yang";
  }

  import ietf-interfaces {
    prefix "if";
  }

  import ietf-routing {
    prefix "rt";
  }

  import ietf-interfaces-common {
    prefix if-cmn;
  }

  import ietf-if-l3-vlan {
    prefix "if-13-vlan";
  }

  organization
    "IETF RTGWG - Routing Area Working Group";
  contact
    "WG Web: <http://tools.ietf.org/wg/rtgwg/>"
    "WG List: <mailto:rtgwg@ietf.org>"
    "Editor: Yingzhen Qu"
    "mailto:yingzhen.qu@huawei.com"
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, and modified across different routing protocols. This module is intended to be used in conjunction with routing protocol configuration modules (e.g., BGP) defined in other models.

Route policy expression:

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

Route policy evaluation:

Policy definitions are referenced in routing protocol configurations using import and export configuration statements. The arguments are members of an ordered list of named policy definitions which comprise a policy chain, and optionally, an explicit default policy action (i.e., reject or accept).

Evaluation of each policy definition proceeds by evaluating its corresponding individual policy statements in order. When a condition statement in a policy statement is satisfied, the corresponding action statement is executed. If the action statement has either accept-route or reject-route actions, policy evaluation of the current policy definition stops, and no further policy definitions in the chain are evaluated.

If the condition is 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).

Policy ‘subroutines’ (or nested policies) are supported by
allowing policy statement conditions to reference another
policy definition which applies conditions and actions from
the referenced policy before returning to the calling policy
statement and resuming evaluation. If the called policy
results in an accept-route (either explicit or by default),
then the subroutine returns an effective true value to the
calling policy. Similarly, a reject-route action returns
false. If the subroutine returns true, the calling policy
continues to evaluate the remaining conditions (using a
modified route if the subroutine performed any changes to the
route).

revision "2019-03-06" {
  description
    "Initial revision.";
  reference
    "RFC XXXX: Routing Policy Configuration Model for Service
Provider Networks";
}

// typedef statements
typedef default-policy-type {
  // this typedef retained for name compatibility with default
  // import and export policy
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";
}
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 2178 - 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; 
}

// grouping statements

grouping prefix-set {
    description
        "Configuration data for prefix sets used in policy
definitions.";

    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";
            }
            enum mixed {
                description
                    "Prefix set contains mixed IPv4 and IPv6 prefixes";
            }
        }
        description
            "Indicates the mode of the prefix set, in terms of which
            address families (IPv4, IPv6, or both) are present. The
            mode provides a hint, but the device must validate that all
            prefixes are of the indicated type, and is expected to
            reject the configuration if there is a discrepancy. The
            MIXED mode may not be supported on devices that require
            prefix sets to be of only one address family.";
    }
}

grouping prefix-set-top {
    description

"Top-level data definitions for a list of IPv4 or IPv6 prefixes which are matched as part of a policy";

container prefix-sets {
  description "Enclosing container ";

  list prefix-set {
    key "name";
    description "List of the defined prefix sets";

    uses prefix-set;

    uses prefix-top;
  }
}

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

  leaf ip-prefix {
    type inet:ip-prefix;
    mandatory true;
    description "The prefix member in CIDR notation -- while the prefix may be either IPv4 or IPv6, most implementations require all members of the prefix set to be the same address family. Mixing address types in the same prefix set is likely to cause an error.";
  }

  leaf masklength-lower {
    type uint8;
    description "Masklength range lower bound.";
  }

  leaf masklength-upper {
    type uint8 {
      range "1..128";
    } must "./masklength-upper >= ../masklength-lower" {
      error-message "The upper bound should not be less than lower bound.";
    }
    description
"Masklength range upper bound.

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

Example: 10.3.192.0/21 through 10.3.192.0/24 would be expressed as prefix: 10.3.192.0/21,
masklength-lower=21,
masklength-upper=24

Example: 10.3.192.0/21 (an exact match) would be expressed as prefix: 10.3.192.0/21,
masklength-lower=21,
masklength-upper=21"
leaf-list address {
    type inet:ip-address;
    description
        "List of IP addresses in the neighbor set";
}

grouping neighbor-set-top {
    description
        "Top-level data definition for a list of IPv4 or IPv6
        neighbors which can be matched in a routing policy";

cointainer neighbor-sets {
    description
        "Enclosing container for the list of neighbor set
        definitions";

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

        uses neighbor-set;
    }
}

} grouping tag-set-top {
    description
        "Top-level data definitions for a list of tags which can
be matched in policies;charset

container tag-sets {
  description
    "Enclosing container for the list of tag sets.";

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

grouping match-set-options-group {
  description
    "Grouping containing options relating to how a particular set
     should 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 inverting the match. Matching on ALL is not supported;

```xml

<grouping name="match-interface-condition">
  <description>This grouping provides interface match condition</description>
  <container name="match-interface">
    <leaf name="interface">
      <type>leafref</type>
      <path>/if:interfaces/if:interface/if:name</path>
      <description>Reference to a base interface. If a reference to a subinterface is required, this leaf must be specified to indicate the base interface.</description>
    </leaf>
    <leaf name="subinterface">
      <type>leafref</type>
      <path>/if:interfaces/if:interface/if-cmn:encapsulation/if-l3-vlan:dot1q-vlan/if-l3-vlan:outer-tag/if-l3-vlan:vlan-id</path>
      <description>Reference to a subinterface -- this requires the base interface to be specified using the interface leaf in this container. If only a reference to a base interface is required, this leaf should not be set.</description>
    </leaf>
  </container>
</grouping>
```

```xml
<grouping name="prefix-set-condition">
  <description>This grouping provides prefix-set conditions</description>
  <container name="match-prefix-set">
    <leaf name="prefix-set">
      <type>leafref</type>
      <path>../../../../../../../defined-sets/</path>
    </leaf>
  </container>
</grouping>
```
"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";
}

}

grouping neighbor-set-condition {
  
  description  
    "This grouping provides neighbor-set conditions";

  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 according to the logic  
      defined in the match-set-options-leaf";
  }
}

}

grouping tag-set-condition {
  
  description  
    "This grouping provides tag-set conditions";

  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-restricted-group;

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


grouping generic-conditions {
  description "Condition statement definitions for checking membership in a generic defined set";

  uses match-interface-condition;
  uses prefix-set-condition;
  uses neighbor-set-condition;
  uses tag-set-condition;
}

grouping policy-conditions {
  description "Data for general policy conditions, i.e., those not related to match-sets";

  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 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 and implementation dependent";
  }

  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;
}
}
grouping policy-conditions-top {
  description
  "Top-level grouping for policy conditions";

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

    uses policy-conditions;
    uses generic-conditions;
  }
}

grouping policy-statements {
  description
  "Data for policy statements";

  leaf name {
    type string;
    description
    "Name of the policy statement";
  }
}

grouping policy-actions {
  description
  "Top-level grouping for policy actions";

  container actions {
    description
    "Top-level container for policy action statements";

    leaf policy-result {
      type policy-result-type;
      description
      "Select the final disposition for the route, either accept or reject.";
    }
    leaf set-metric {
      type uint32;
      description
      "Set a new metric for the route.";
    }
  }
}
leaf set-preference {
  type uint16;
  description
    "Set a new preference for the route.";
}

grouping policy-statements-top {
  description
    "Top-level grouping for the policy statements list";

  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 (see the description of policy
         evaluation at the top of this module.";

      uses policy-statements;

      uses policy-conditions-top;
      uses policy-actions;
    }
  }
}

grouping policy-definitions {
  description
    "This grouping provides policy definitions";

  leaf name {
    type string;
    description
      "Name of the top-level policy definition -- this name
       is used in references to the current policy";
  }
}

grouping apply-policy-import {

description
"Grouping for applying import policies";

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 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.";
}

grouping apply-policy-export {
    description
    "Grouping for applying export policies";

    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
        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.";
}
}

grouping apply-policy {
  description
  "Configuration data for routing policies";
  uses apply-policy-import;
  uses apply-policy-export;

  container apply-policy-state {
    description
    "Operational state associated with routing policy";
    //TODO: identify additional state data beyond the intended
    //policy configuration.
  }
}

container 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.";

    uses apply-policy;
  }
}

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

  container defined-sets {

11. Policy examples

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

<?yfile include="file:///tmp/routing-policy-example-draft.xml"?>

12. References
12.1. Normative references

[I-D.ietf-netmod-intf-ext-yang]

[I-D.ietf-netmod-sub-intf-vlan-model]


12.2. Informative references


Appendix A. Acknowledgements

The routing policy module defined in this draft 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.

Authors’ Addresses

Yingzhen Qu
Futurewei
2330 Central Expressway
Santa Clara CA 95050
USA

Email: yingzhen.qu@futurewei.com
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 IGP network. 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, dramatically reducing the operational need to control the tie-breaks among various FRR options.
1. Introduction .................................................. 3
  1.1. Conventions used in this document ....................... 7
2. Terminology .................................................. 7
3. Intersecting P-Space and Q-Space with post-convergence paths 8
  3.1. P-Space property computation for a resource X ........... 8
  3.2. Q-Space property computation for a link S-F, over post-
       convergence paths ....................................... 8
  3.3. Q-Space property computation for a set of links adjacent
       to S, over post-convergence paths ....................... 9
  3.4. Q-Space property computation for a node F, over post-
       convergence paths ....................................... 9
  3.5. Scaling considerations when computing Q-Space ........... 9
4. TI-LFA Repair Tunnel ......................................... 9
  4.1. FRR path using a direct neighbor ........................ 10
  4.2. FRR path using a PQ node ................................ 10
  4.3. FRR path using a P node and Q node that are adjacent ... 10
  4.4. Connecting distant P and Q nodes along post-convergence
       paths ...................................................... 10
5. Protecting segments ......................................... 10
  5.1. The active segment is a node segment .................... 11
  5.2. The active segment is an adjacency segment ............... 11
      5.2.1. Protecting [Adjacency, Adjacency] segment lists ... 11
      5.2.2. Protecting [Adjacency, Node] segment lists ........ 12
  5.3. Protecting SR policy midpoints against node failure .... 13
      5.3.1. Protecting (F, T, D) or (S->F, T, D) ............... 13
      5.3.2. Protecting (F, F->T, D) or (S->F, F->T, D) .......... 14
6. TI-LFA and SR algorithms .................................. 15
7. Usage of Adjacency segments in the repair list ............. 16
8. Measurements on Real Networks ............................... 16
1. 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 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 be applicable in some topologies ([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 routing domain. 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 RLFAs or DLFAs, 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 only 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).
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 of 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 6.

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, in an MPLS world, this may create a long stack of labels to be pushed 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 path 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](image)

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

Section 3 suggests to compute the P-Space and Q-Space properties defined in Section 2, for the specific case of nodes lying over the post-convergence paths towards the protected destinations.

Using the properties defined in Section 3, Section 4 describes how to compute protection lists that encode a loop-free post-convergence path towards the destination.

Section 5 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 8 summarizes these measurements.

1.1. 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.

2. 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 node \( R \) w.r.t. a resource \( X \) (e.g. a link \( S-F \), a node \( F \), or a SRLG) is the set of nodes that are reachable from \( R \) without passing through \( X \). It is the set of nodes that are not downstream of \( X \) in \( SPT_{old}(R) \).

The Extended P-Space \( P'(R, X) \) of a node \( R \) w.r.t. a resource \( X \) is the set of nodes that are reachable from \( R \) or a neighbor of \( R \), without passing through \( X \).

The Q-Space \( Q(D, X) \) of a destination node \( D \) w.r.t. a resource \( X \) is the set of nodes which do not use \( X \) to reach \( D \) in the initial state of the network. In other words, it is the set of nodes which have \( D \) in their P-Space w.r.t. \( S-F, F, \) or a set of links adjacent to \( S \).

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

3. 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 4 to compute the TI-LFA repair list.

3.1. P-Space property computation for a resource \( X \)

A node \( N \) is in \( P(R, X) \) if it is not downstream of \( X \) in \( SPT_{old}(R) \). \( X \) can be a link, a node, or a set of links adjacent to the PLR. A node \( N \) is in \( P'(R, X) \) if it is not downstream of \( X \) in \( SPT_{old}(N) \), for at least one neighbor \( N \) of \( R \).

3.2. Q-Space property computation for a link \( S-F \), over post-convergence paths

We want to determine which nodes on the post-convergence path from the PLR to the destination \( D \) are in the Q-Space of destination \( D \) w.r.t. link \( S-F \).
This can be found by intersecting the post-convergence path to D, assuming the failure of S-F, with Q(D, S-F).

3.3. Q-Space property computation for a set of links adjacent to S, over post-convergence paths

We want to determine which nodes on the post-convergence path from the PLR to the destination D are in the Q-Space of destination D w.r.t. a set of links adjacent to S (S being the PLR). That is, we aim to find the set of nodes on the post-convergence path that use none of the members of the protected set of links, to reach D.

This can be found by intersecting the post-convergence path to D, assuming the failure of the set of links, with the intersection among Q(D, S->X) for all S->X belonging to the set of links.

3.4. Q-Space property computation for a node F, over post-convergence paths

We want to determine which nodes on the post-convergence from the PLR to the destination D are in the Q-Space of destination D w.r.t. node F.

This can be found by intersecting the post-convergence path to D, assuming the failure of F, with Q(D, F).

3.5. 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 SPT 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.

4. TI-LFA Repair Tunnel

The TI-LFA repair tunnel consists of an outgoing interface and a list of segments (repair list) 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 tunnel 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 8 provides statistics for the number of SIDs in the explicit path to protect against various failures.

4.1. 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 MUST be empty.

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

4.2. 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 MUST be set to the outgoing interface used to reach R.

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

4.3. 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 MUST 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 repair tunnel.

4.4. 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. 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.

5. Protecting segments

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-P.
The behavior depends on the type of active segment to be protected.

5.1. The active segment is a node segment

The active segment MUST be kept on the SR header unchanged and the repair list MUST be inserted at the head of the list. The active segment becomes the first segment of the inserted repair list.

This behavior is slightly modified when SR-MPLS is used:

- If the repair list ends with an adjacency segment terminating on the tail-end of the active segment, and if the active segment has been signalled with penultimate hop popping, the active segment MUST be popped before pushing the repair list.

- If the SRGB at the Q node is different from the SRGB at the PLR, then the active segment (before the insertion of the repair list) MUST be updated to fit the SRGB of the Q node.

In Section 5.3, we describe the node protection behavior of PLR S, for the specific case where the active segment is a prefix segment for the neighbor F itself.

5.2. 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 signalled in the IGP, any SR policy using this segment knows that it may be transiently rerouted out of S-F in case of S-F failure.

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.

5.2.1. 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 two consecutive "PUSH" operations: first it pushes a node segment for F, and then it pushes a repair list allowing to reach F while bypassing S-F. 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 \([\text{RT}(F), \text{node}(F), \text{adj}(F-M)]\), where \(\text{RT}(F)\) is the repair tunnel for destination \(F\).

This behavior is slightly modified when SR-MPLS is used:

- If the repair list ends with an adjacency segment terminating on \(F\), and if the node segment of \(F\) has been signalled with penultimate hop popping, the implementation MUST pop \(\text{Adj}(S-F)\) and then push the repair list (the node segment of \(F\) is not pushed). The packet will leave \(S\) with a segment list matching \([\text{RT}(F), \text{adj}(F-M)]\).

- If the SRGB at the Q node is different from the SRGB at the PLR, then MPLS label representing node\((F)\) MUST be calculated as per the SRGB of the Q node.

In Section 5.3.2, we describe the TI-LFA behavior of PLR \(S\) when node protection is applied and the two first segments are Adjacency Segments.

5.2.2. Protecting \([\text{Adjacency}, \text{Node}]\) segment lists

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

A first solution would consist in steering the packet back to \(F\) while avoiding \(S-F\). To do so, \(S\) MUST apply a "NEXT" operation on \(\text{Adj}(S-F)\) and then two consecutive "PUSH" operations: first it pushes a node segment for \(F\), and then it pushes a repair list allowing to reach \(F\) while bypassing \(S-F\).

Upon failure of \(S-F\), a packet reaching \(S\) with a segment list matching \([\text{adj}(S-F), \text{node}(T), ...]\) will thus leave \(S\) with a segment list matching \([\text{RT}(F), \text{node}(F), \text{node}(T)]\).

This behavior is slightly modified when SR-MPLS is used:

- If the repair list ends with an adjacency segment terminating on \(F\), and if the node segment of \(F\) has been signalled with penultimate hop popping, the implementation MUST pop \(\text{Adj}(S-F)\) and then push the repair list (the node segment of \(F\) is not pushed). The packet will leave \(S\) with a segment list matching \([\text{RT}(F), \text{node}(T)]\).

- If the SRGB at the Q node is different from the SRGB at the PLR, then MPLS label representing node\((F)\) MUST be calculated as per the SRGB of the Q node.
Another solution is to not steer the packet back via F but rather follow the new shortest path to 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 [RT(Q), node(T), ...]. Note that this second behavior is the one followed for node protection, as described in Section 5.3.1.

This behavior is slightly modified when SR-MPLS is used:

- If the repair list ends with an adjacency segment terminating on T (T being the Q node), and if the node segment of T has been signalled with penultimate hop popping, the implementation MUST pop Adj(S-F) and then push the repair list (the node segment of T is not pushed). The packet will leave S with a segment list matching [RT(Q=T), ...].

- If the SRGB at the Q node is different from the SRGB at the PLR, then the MPLS label representing node(T) MUST be calculated as per the SRGB of the Q node.

The first proposal 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 existing path. As stated in Section 1, 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.

5.3. Protecting SR policy midpoints against node failure

In this section, we describe the behavior of a node S configured to interpret the failure of link S->F as the node failure of F, in the specific case where the active segment of the packet received by S is a Prefix SID of F represented as "F"), or an Adjacency SID for the link S-F (represented as "S->F").

5.3.1. Protecting (F, T, D) or (S->F, T, D)

This section describes the protection behavior of S when all of the following conditions are true:

1. the active segment is a prefix SID for a neighbor F, or an adjacency segment S->F

2. the primary interface used to forward the packet failed
3. the segment following the active segment is a prefix SID (for node T)
4. node protection is active for that interface.

In such a case, the PLR MUST:

1. apply a NEXT operation; the segment F or S->F is removed
2. Confirm that the next segment is in the SRGB of F, meaning that the next segment is a prefix segment, e.g. for node T
3. Retrieve the segment ID of T (as per the SRGB of F)
4. Apply a NEXT operation followed by a PUSH operation of T’s segment based on the SRGB of node S.
5. Look up T’s segment (based on the updated label value) and forward accordingly.

5.3.2. Protecting {F, F->T, D} or {S->F, F->T, D}

This section describes the protection behavior of S when all of the following conditions are true:

1. the active segment is a prefix SID for a neighbor F, or an adjacency segment S->F
2. the primary interface used to forward the packet failed
3. the segment following the active segment is an adjacency SID (F->T)
4. node protection is active for that interface.

In such a case, the PLR MUST:

1. Apply a NEXT operation; the segment F or S->F is removed
2. Confirm that the next segment is an adjacency SID of F, say F->T
3. Retrieve the node segment ID associated to T (as per the set of Adjacency Segments of F)
4. Apply a NEXT operation on the next segment followed by a PUSH of T’s segment based on the SRGB of the node S.
5.  Look up T’s segment (based on the updated label value) and forward accordingly.

It is noteworthy to mention that node "S" in the procedures described in Sections 5.3.1 and 5.3.2 can always determine whether the segment after popping the top segment is an adjacency SID or a prefix-SID of the next-hop "F" as follows:

1.  In a link state environment, the node "S" knows the SRGB and the adj-SIDs of the neighboring node "F"

2.  If the new segment after popping the top segment is within the SRGB or the adj-SIDs of "F", then node "S" is certain that the failure of node "F" is a midpoint failure and hence node "S" applies the procedures specified in Sections 5.3.1 or 5.3.2, respectively.

3.  Otherwise the failure is not a midpoint failure and hence the node "S" may apply other protection techniques that are beyond the scope of this document or simply drop the packet and wait for normal protocol convergence.

6.  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 has 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.

When a Node-SID is associated with the SR default algorithm, enforcing TI-LFA to use Node-SIDs associated with a strict SPF algorithm is a definitive solution to this problem.

[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 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.

7. 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.

```
S --- R2 --- R3 --- R4 --- R5 --- D
    \   \   /
     R7 -- R8
    |   |
     R9 -- R10
```

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 \[\text{adj}(R7-R8),\text{adj}(R8-R4)\] 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 \text{adj}(R7-R8), R7 will push an additional repair list onto the packet following the procedures defined in Section 5.

To avoid the possibility of this double FRR, an implementation of TI-LFA MAY pick only non protected adjacency segments when building the repair list.

8. Measurements on Real Networks

This section presents measurements performed on real service provider and large enterprise networks. The objective of the measurements is to assess the number of SIDs required in an explicit path when the mechanism 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 measurements 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 measurements are carried out as follows:

- For each network, the algorithms described in this document are applied to protect all prefixes against link, node, and local SRLG failure.
- For each prefix, the number of SIDs used by the repair path is recorded.
- 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.

<table>
<thead>
<tr>
<th>Network</th>
<th>Nodes</th>
<th>Circuits</th>
<th>Node-to-Link Ratio</th>
<th>SRLG info?</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>408</td>
<td>665</td>
<td>1 : 63</td>
<td>Yes</td>
</tr>
<tr>
<td>T2</td>
<td>587</td>
<td>1083</td>
<td>1 : 84</td>
<td>No</td>
</tr>
<tr>
<td>T3</td>
<td>93</td>
<td>401</td>
<td>4 : 31</td>
<td>Yes</td>
</tr>
<tr>
<td>T4</td>
<td>247</td>
<td>393</td>
<td>1 : 59</td>
<td>Yes</td>
</tr>
<tr>
<td>T5</td>
<td>34</td>
<td>96</td>
<td>2 : 82</td>
<td>Yes</td>
</tr>
<tr>
<td>T6</td>
<td>50</td>
<td>78</td>
<td>1 : 56</td>
<td>No</td>
</tr>
<tr>
<td>T7</td>
<td>82</td>
<td>293</td>
<td>3 : 57</td>
<td>No</td>
</tr>
<tr>
<td>T8</td>
<td>35</td>
<td>41</td>
<td>1 : 17</td>
<td>Yes</td>
</tr>
<tr>
<td>T9</td>
<td>177</td>
<td>1371</td>
<td>7 : 74</td>
<td>Yes</td>
</tr>
</tbody>
</table>

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:

- 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 4.1.

- 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 4.2.

- 2 or more SIDs: The repair path consists of 2 or more SIDs as described in Sections 4.3 and 4.4. We do not cover the case for 2 SIDs (Section 4.3) 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.

<table>
<thead>
<tr>
<th>Network</th>
<th>0 SIDs</th>
<th>1 SID</th>
<th>2 SIDs</th>
<th>3 SIDs</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>74.227%</td>
<td>25.256%</td>
<td>0.517%</td>
<td>0.001%</td>
</tr>
<tr>
<td>T2</td>
<td>81.097%</td>
<td>18.738%</td>
<td>0.165%</td>
<td>0.0%</td>
</tr>
<tr>
<td>T3</td>
<td>95.878%</td>
<td>4.067%</td>
<td>0.056%</td>
<td>0.0%</td>
</tr>
<tr>
<td>T4</td>
<td>62.547%</td>
<td>35.666%</td>
<td>1.788%</td>
<td>0.0%</td>
</tr>
<tr>
<td>T5</td>
<td>85.733%</td>
<td>14.267%</td>
<td>0.0%</td>
<td>0.0%</td>
</tr>
<tr>
<td>T6</td>
<td>81.252%</td>
<td>18.714%</td>
<td>0.033%</td>
<td>0.0%</td>
</tr>
<tr>
<td>T7</td>
<td>98.857%</td>
<td>1.143%</td>
<td>0.0%</td>
<td>0.0%</td>
</tr>
<tr>
<td>T8</td>
<td>94.118%</td>
<td>5.882%</td>
<td>0.0%</td>
<td>0.0%</td>
</tr>
<tr>
<td>T9</td>
<td>98.950%</td>
<td>1.050%</td>
<td>0.0%</td>
<td>0.0%</td>
</tr>
</tbody>
</table>

Table 2A: Link protection (repair size distribution)
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.

<table>
<thead>
<tr>
<th>Network</th>
<th>0 SIDs</th>
<th>1 SID</th>
<th>2 SIDs</th>
<th>3 SIDs</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>74.177%</td>
<td>25.306%</td>
<td>0.517%</td>
<td>0.001%</td>
</tr>
<tr>
<td>T2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T3</td>
<td>93.650%</td>
<td>6.301%</td>
<td>0.049%</td>
<td>0.0%</td>
</tr>
<tr>
<td>T4</td>
<td>62.547%</td>
<td>35.666%</td>
<td>1.788%</td>
<td>0.0%</td>
</tr>
<tr>
<td>T5</td>
<td>83.139%</td>
<td>16.861%</td>
<td>0.0%</td>
<td>0.0%</td>
</tr>
<tr>
<td>T6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T7</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T8</td>
<td>85.185%</td>
<td>14.815%</td>
<td>0.0%</td>
<td>0.0%</td>
</tr>
<tr>
<td>T9</td>
<td>98.940%</td>
<td>1.060%</td>
<td>0.0%</td>
<td>0.0%</td>
</tr>
</tbody>
</table>

Table 3A: Local SRLG protection repair size distribution

<table>
<thead>
<tr>
<th>Network</th>
<th>0 SIDs</th>
<th>1 SID</th>
<th>2 SIDs</th>
<th>3 SIDs</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>74.177%</td>
<td>99.482%</td>
<td>99.999%</td>
<td>100.001%</td>
</tr>
<tr>
<td>T2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T3</td>
<td>93.650%</td>
<td>99.951%</td>
<td>100.000%</td>
<td>0.0%</td>
</tr>
</tbody>
</table>
The remaining two tables summarize the measurements on the number of SIDs needed for node protection.

**Table 3B: Local SRLG protection repair size Cumulative distribution**

<table>
<thead>
<tr>
<th></th>
<th>0 SIDs</th>
<th>1 SID</th>
<th>2 SIDs</th>
<th>3 SIDs</th>
<th>4 SIDs</th>
</tr>
</thead>
<tbody>
<tr>
<td>T4</td>
<td>62.547%</td>
<td>98.212%</td>
<td>100.000%</td>
<td>100.0%</td>
<td></td>
</tr>
<tr>
<td>T5</td>
<td>83.139%</td>
<td>100.000%</td>
<td>100.0%</td>
<td>100.0%</td>
<td></td>
</tr>
<tr>
<td>T6</td>
<td>No SRLG Information</td>
<td>No SRLG Information</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T7</td>
<td>No SRLG Information</td>
<td>No SRLG Information</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T8</td>
<td>85.185%</td>
<td>100.000%</td>
<td>100.000%</td>
<td>100.0%</td>
<td></td>
</tr>
<tr>
<td>T9</td>
<td>98.940%</td>
<td>100.000%</td>
<td>100.000%</td>
<td>100.0%</td>
<td></td>
</tr>
</tbody>
</table>

**Table 4A: Node protection (repair size distribution)**

<table>
<thead>
<tr>
<th></th>
<th>0 SIDs</th>
<th>1 SID</th>
<th>2 SIDs</th>
<th>3 SIDs</th>
<th>4 SIDs</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>49.771%</td>
<td>47.902%</td>
<td>2.156%</td>
<td>0.148%</td>
<td>0.023%</td>
</tr>
<tr>
<td>T2</td>
<td>36.528%</td>
<td>59.625%</td>
<td>3.628%</td>
<td>0.194%</td>
<td>0.025%</td>
</tr>
<tr>
<td>T3</td>
<td>73.287%</td>
<td>25.574%</td>
<td>1.128%</td>
<td>0.010%</td>
<td>0%</td>
</tr>
<tr>
<td>T4</td>
<td>36.112%</td>
<td>57.350%</td>
<td>6.329%</td>
<td>0.199%</td>
<td>0.010%</td>
</tr>
<tr>
<td>T5</td>
<td>73.185%</td>
<td>26.815%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>T6</td>
<td>78.362%</td>
<td>21.320%</td>
<td>0.318%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>T7</td>
<td>66.106%</td>
<td>32.813%</td>
<td>1.082%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>T8</td>
<td>59.712%</td>
<td>40.288%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>T9</td>
<td>98.950%</td>
<td>1.050%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
</tbody>
</table>
Table 4B: Node protection (repair size cumulative distribution)

| T4 | 36.112% | 93.461% | 99.791% | 99.990% | 100% |
| T5 | 73.185% | 100.0%  | 100.0%  | 100.0%  | 100% |
| T6 | 78.362% | 99.682% | 100.0%  | 100.0%  | 100% |
| T7 | 66.106% | 98.918% | 100.0%  | 100.0%  | 100% |
| T8 | 59.712% | 100.0%  | 100.0%  | 100.0%  | 100% |
| T9 | 98.950% | 100.0%  | 100.0%  | 100.0%  | 100% |

9. 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 do not protect external destinations, but rather provides quick restoration for destination that are internal to a routing domain.

10. IANA Considerations

No requirements for IANA

11. Conclusions

This document proposes a mechanism that is able to pre-calculate a backup path for every primary path so as to be able to protect against the failure of a directly connected link, node, or SRLG. The mechanism is able to calculate the backup path irrespective of the topology as long as the topology is sufficiently redundant.

12. Acknowledgments

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

13.1. Normative References


13.2. Informative References


Authors' Addresses

Stephane Litkowski
Orange
France

Email: stephane.litkowski@orange.com

Ahmed Bashandy
Individual

Email: abashandy.ietf@gmail.com

Clarence Filsfils
Cisco Systems
Brussels
Belgium

Email: cfilsfil@cisco.com

Bruno Decraene
Orange
Issy-les-Moulineaux
France

Email: bruno.decrane@orange.com
Pierre Francois
INSA Lyon
Email: pierre.francois@insa-lyon.fr

Daniel Voyer
Bell Canada
Canada
Email: daniel.voyer@bell.ca

Francois Clad
Cisco Systems
Email: fclad@cisco.com

Pablo Camarillo
Cisco Systems
Email: pcamaril@cisco.com
RIB YANG Data Model
draft-ietf-rtgwg-yang-rib-extend-02

Abstract

The 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 March 11, 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
1. Introduction

This document defines a YANG, [RFC6020][RFC7950], data model which extends the generic data model for RIB by augmenting the ietf-routing model as defined in [RFC8349].

RIB is a collection of best routes from all routing protocols. Within a protocol routes are selected based on the metrics in use by that protocol, and the protocol install its best routes to RIB. RIB selects the best route by comparing the route preference (aka, administrative distance) of the associated protocol.

The augmentations described herein extend the RIB to support multiple paths per route, 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]:

- client
- server
- configuration
- system state
- operational state
- intended configuration

The following terms are defined in [RFC7950]:

- action
- augment
- container
- container with presence
- data model
- data node
- feature
- leaf
- list
- mandatory node
- module
- schema tree
2.1. Glossary of New Terms

Routing Information Base (RIB): An object containing a list of routes, together with other information. See [RFC8349] Section 5.2 for details.

2.2. Tree Diagrams

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

2.3. 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.

<table>
<thead>
<tr>
<th>Prefix</th>
<th>YANG module</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>if</td>
<td>ietf-interfaces</td>
<td>[RFC8343]</td>
</tr>
<tr>
<td>rt</td>
<td>ietf-routing</td>
<td>[RFC8349]</td>
</tr>
<tr>
<td>v4ur</td>
<td>ietf-ipv4-unicast-routing</td>
<td>[RFC8349]</td>
</tr>
<tr>
<td>v6ur</td>
<td>ietf-ipv6-unicast-routing</td>
<td>[RFC8349]</td>
</tr>
<tr>
<td>inet</td>
<td>ietf-inet-types</td>
<td>[RFC6991]</td>
</tr>
</tbody>
</table>

Table 1: Prefixes and Corresponding YANG Modules

3. Design of the Model

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

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

Individual routes tags will be 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.

3.2. Multiple next-hops

Both IPv4 and IPv6 static route configuration defined in [RFC8349] have been augmented with a multi-next-hop option.

A static route/prefix can be configured to have multiple next-hops, each with their own tag and route preference.

In RIB, a route may have multiple next-hops. They can be either equal cost multiple paths (ECMP), or they may have different metrics.

3.3. Repair path

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

A repair path is augmented in RIB operation state for each path.

4. RIB Model Tree

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

```
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
        ++--rw application-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
        ++--rw application-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
  +--rw application-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
  +--rw application-tag? uint32
augment /rt:routing/rt:ribs/rt:rib:
  +--ro rib-summary-statistics
      +--ro total-routes?  uint32
      +--ro total-active-routes?  uint32
      +--ro total-route-memory?   uint64
      +--ro protocol-rib-statistics*  []
          +--ro rib-protocol?  identityref
          +--ro protocol-total-routes?  uint32
          +--ro protocol-active-routes?  uint32
          +--ro protocol-route-memory?   uint64
augment /rt:routing/rt:ribs/rt:rib/rt:routes/rt:route:
  +--ro metric?          uint32
  +--ro tag?             uint32
  +--ro application-tag? uint32
augment /rt:routing/rt:ribs/rt:rib/rt:routes:
  +--ro repair-route* [id]
      +--ro id       string
      |   +--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:simple-next-hop:
  +--ro repair-path?
      -> /rt:routing/ribs/rib/routes/repair-route/id
augment /rt:routing/rt:ribs/rt:rib/rt:routes/rt:route
         /rt:next-hop/rt:next-hop-options/rt:special-next-hop:
  +--ro repair-path?
      -> /rt:routing/ribs/rib/routes/repair-route/id
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?
      -> /rt:routing/ribs/rib/routes/repair-route/id
5. RIB YANG Model

<CODE BEGINS>
module ietf-rib-extension {
  yang-version "1.1";

  prefix rib-ext;

  import ietf-inet-types {
    prefix "inet";
  }

  import ietf-interfaces {
    prefix "if";
  }

  import ietf-routing {
    prefix "rt";
  }

  import ietf-ipv4-unicast-routing {
    prefix "v4ur";
  }

  import ietf-ipv6-unicast-routing {
    prefix "v6ur";
  }

  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@huawei.com>"

  description
    "This YANG module extends the generic data model for RIB by augmenting the ietf-netmod-routing-cfg model. It is intended that the module will be extended by vendors to define vendor-specific RIB parameters."

    This YANG model conforms to the Network Management
Datastore Architecture (NDMA) as described in RFC 8242.

Copyright (c) 2018 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 2019-03-11 {
  description
    "Initial RFC Version";
  reference
    "RFC XXXX: A YANG Data Model for RIB Extensions.";
}

/* Groupings */
grouping rib-statistics {
  description "Statistics grouping used for RIB augmentation";
  container rib-summary-statistics {
    config false;
    description "Container for RIB statistics";
    leaf total-routes {
      type uint32;
      description "Total routes in the RIB from all protocols";
    }
    leaf total-active-routes {
      type uint32;
      description "Total active routes in the RIB";
    }
    leaf total-route-memory {
      type uint64;
      description "Total memory for all routes in the RIB from all protocol clients";
    }
    list protocol-rib-statistics {
      description "List protocol statistics";
      leaf rib-protocol {
        type identityref {
          description "Protocol identifier";
        }
      }
    }
  }
}
base rt:routing-protocol;
)
description "Routing protocol for statistics";
)
leaf protocol-total-routes {
    type uint32;
    description
        "Total number routes for protocol in the RIB";
}
leaf protocol-active-routes {
    type uint32;
    description
        "Number active routes for protocol in the RIB";
}
leaf protocol-route-memory {
    type uint64;
    description
        "Total memory for all routes in the RIB for protocol";
}
}
}
)

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 paths";
    leaf metric {
        type uint32;
        description "Route metric";
    }
    leaf tag {
        type uint32;
        description "Route tag";
    }
leaf application-tag {
  type uint32;
  description "Additional Application-Specific Route tag. This additional tag can be used by applications that require semantics and 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 unless done so explicitly.";
}

grouping path-attribute {
  description
    "Path attribute grouping";
  leaf repair-path {
    type leafref {
      path "/rt:routing/rt:ribs/rt:rib/"
      + "rt:routes/repair-route/id";
    }
    description
      "IP Fast ReRoute (IPFRR) repair path, use a path from repair-route list";
    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 "Route preference - Used to select among multiple static routes with a lower preference next-hop preferred and equal preference paths yielding Equal Cost Multi-Path (ECMP).";
    }
    leaf tag {
      type uint32;
      default "0";
      description "Route tag";
    }
  }
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 "Route preference - Used to select among multiple
static routes with a lower preference next-hop
preferred and equal preference paths yielding
Equal Cost Multi-Path (ECMP)."
  }

  leaf tag {
    type uint32;
    default "0";
    description "Route tag"
  }
}

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 "Route preference - Used to select among multiple
static routes with a lower preference next-hop
preferred and equal preference paths yielding
Equal Cost Multi-Path (ECMP)."
  }

  leaf tag {
    type uint32;
    default "0";
    description "Route tag"
  }
}
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 "Route preference - Used to select among multiple
    static routes with a lower preference next-hop
    preferred and equal preference paths yielding
    Equal Cost Multi-Path (ECMP).";
  }
  leaf tag {
    type uint32;
    default "0";
    description "Route tag";
  }
}

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 tag.";
  uses attributes;
}

augment "/rt:routing/rt:ribs/rt:rib/"
  + "rt:routes"
{
  description
  "Augment a route with a list of repair-paths.";
  list repair-route {
    key "id";
    description
    "A repair-path entry, which can be referenced
    by a repair-path.";
    leaf id {

type string;
description
  "A unique identifier."
}

container next-hop {
description
  "Route's next-hop attribute."
  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 "Route metric"
  }
}

augment "/rt:routing/rt:ribs/rt:rib/"
  + "rt:routes/rt:route/rt:next-hop/rt:next-hop-options/"
  + "rt:simple-next-hop"
{
description
  "Add more parameters to a path."
  uses path-attribute;
}

augment "/rt:routing/rt:ribs/rt:rib/"
  + "rt:routes/rt:route/rt:next-hop/rt:next-hop-options/"
  + "rt:special-next-hop"
{
description
  "Add more parameters to a path."
  uses path-attribute;
}

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
    "This case augments the 'next-hop-options' in the routing model."
    uses path-attribute;
} }

6. Security Considerations

The YANG modules specified in this document define 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 [RFC5246].

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. For these augmentations to ietf-routing.yang, the ability to delete, add, and modify IPv4 and IPv6 static routes 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. 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 [XML-REGISTRY]. Following the format in [RFC3688], the following registration is requested to be made:


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].


8. References

8.1. Normative References


Lindem & Qu Expires March 11, 2020 [Page 15]


8.2. Informative References


Appendix A. Combined Tree Diagram

This appendix includes the combined ietf-routing.yang and ietf-rib-extensions.yang tree diagram.

```yang
module: ietf-routing
  +++rw routing
  | +++rw router-id? yang:dotted-quad
  | +++ro interfaces
  | | +++ro interface* if:interface-ref
  | +++rw control-plane-protocols
  | | +++rw control-plane-protocol* [type name]
  | | | +++rw type identityref
  | | | +++rw name identityref
  | | | +++rw description? string
  | | +++rw static-routes
  | +---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
  | | | | | | +---:(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 source-protocol identityref
  | | | | +---ro active? empty
  | | | | +---ro last-updated? yang:date-and-time
```

Lindem & Qu Expires March 11, 2020 [Page 17]
module: ietf-rib-extension
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
            +--rw application-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
            +--rw application-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
            +--rw application-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
            +--rw application-tag?   uint32
---rw application-tag?   uint32
augment /rt:routing/rt:ribs/rt:rib:
  +--ro rib-summary-statistics
    +--ro total-routes?         uint32
    +--ro total-active-routes?  uint32
    +--ro total-route-memory?   uint64
    +--ro protocol-rib-statistics* []
      +--ro rib-protocol?       identityref
      +--ro protocol-total-routes? uint32
      +--ro protocol-active-routes? uint32
      +--ro protocol-route-memory? uint64
augment /rt:routing/rt:ribs/rt:rib/rt:routes/rt:route:
  +--ro metric?            uint32
  +--ro tag?               uint32
  +--ro application-tag?   uint32
augment /rt:routing/rt:ribs/rt:rib/rt:routes:
  +--ro repair-route* [id]
    +--ro id          string
    +--ro next-hop
      |  +--ro outgoing-interface? if:interface-state-ref
      |  +--ro next-hop-address? inet:ip-address
    +--ro metric?     uint32
   /rt:next-hop-options/rt:simple-next-hop:
   /rt:next-hop-options/rt:special-next-hop:
   /rt:next-hop-options/rt:next-hop-list/rt:next-hop:

Appendix B.  ietf-rib-extension.yang examples

Examples will be added in a future version of this document.

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 and Rob Wilton for review and
comments.

Authors’ Addresses
Acee Lindem
Cisco Systems
301 Midenhall Way
Cary, NC  27513
USA

EMail: acee@cisco.com

Yingzhen Qu
Huawei
2330 Central Expressway
Santa Clara, CA  95050
USA

EMail: yingzhen.qu@huawei.com
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.
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:

- 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.

- 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.

- 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
### Option Type: TBD_0

Opt Data Len: 8 octets.

Other fields: refer to [RFC8300].

#### 2. IOAM Service Option

<table>
<thead>
<tr>
<th>0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>+---------------------------------------------+</td>
</tr>
<tr>
<td>Option Type</td>
</tr>
<tr>
<td>-----------------------------------------------</td>
</tr>
<tr>
<td>Ver</td>
</tr>
<tr>
<td>-----------------------------------------------</td>
</tr>
<tr>
<td>Service Path Identifier</td>
</tr>
</tbody>
</table>

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
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

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.

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SFC Type</td>
<td>Length</td>
<td>Reserved</td>
</tr>
<tr>
<td>+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SFC Metadata Class</td>
<td>Type</td>
<td>U</td>
</tr>
<tr>
<td>+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Variable-Length Metadata</td>
<td></td>
<td></td>
</tr>
<tr>
<td>+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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.

```
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   IOAM Type   |     Length    |            Reserved           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            IOAM Service Metadata Options (variable)           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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.
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

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBD_0</td>
<td>NSH Service Option</td>
<td>[This draft]</td>
</tr>
<tr>
<td>TBD_1</td>
<td>IOAM Service Option</td>
<td>[This draft]</td>
</tr>
<tr>
<td>TBD_2</td>
<td>PBT Service Option</td>
<td>[This draft]</td>
</tr>
<tr>
<td>TBD_3</td>
<td>IFA Service Option</td>
<td>[This draft]</td>
</tr>
<tr>
<td>TBD_4</td>
<td>SFC Service Metadata Type</td>
<td>[This draft]</td>
</tr>
<tr>
<td>TBD_5</td>
<td>IOAM Service Metadata Type</td>
<td>[This draft]</td>
</tr>
<tr>
<td>TBD_6</td>
<td>IFA Service Metadata Type</td>
<td>[This draft]</td>
</tr>
</tbody>
</table>

6. Security Considerations

TBD.

7. References

7.1. Normative References

Li, et al. Expires March 13, 2020
Internet-Draft        IPv6 Encaps for SFC and IFIT        September 2019

[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,
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,


7.2. Informative References


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

1. Introduction............................................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:

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

```yang
++--rw tnlmGlobal
  ++--rw nonexistentCheckFlag?  boolean
  ++--rw tunnelPolicys
    ++--rw tunnelPolicy* [tnlPolicyName]
      ++--rw tnlPolicyName     string
      ++--ro tnlPolicyExist?    tnlPolicyExist
      ++--ro tpSubCount?        uint32
      ++--rw description?       string
      ++--rw tnlPolicyType?     tnlmbaseTnlPolicyType
      ++--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?     tnlmbaseSelTnlType
                ++--rw selTnlType2?     tnlmbaseSelTnlType
                ++--rw selTnlType3?     tnlmbaseSelTnlType
                ++--rw selTnlType4?     tnlmbaseSelTnlType
                ++--rw selTnlType5?     tnlmbaseSelTnlType
                ++--rw selTnlType6?     tnlmbaseSelTnlType
                ++--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.

```yang
++-rw tunnelSelectors
  +++-rw tunnelSelector* [name]
    +++-rw name                  string
  +++-rw tunnelSelectorNodes
    +++-rw tunnelSelectorNode* [nodeSequence]
      +++-rw nodeSequence         uint32
      +++-rw matchMode            rtpMatchMode
    +++-rw matchCondition
      +++-rw matchDestPrefixFilters
        +++-rw prefixName?         string
      +++-rw matchIPv4NextHops
        +++-rw prefixName?         string
        +++-rw aclNameOrNum?        string
      +++-rw matchIPv6NextHops
        +++-rw ipv6PrefixName?      string
      +++-rw matchCommunityFilters
        +++-rw communityNameOrNum* [cmntyNameOrNum]
          +++-rw cmntyNameOrNum     string
          +++-rw wholeMatch?         boolean
          +++-rw sortMatch?          boolean
      +++-rw matchRdFilters
        +++-rw rdIndex?             uint32
      +++-rw applyAction
      +++-rw applyTnlPolicys
        +++-rw applyTnlPolicy!     string
          +++-rw tnlPolicyName?      string
```

augment /bgp:bgp/bgpglobal/bgpgafi-safis/bgpgafi-safi/
  bgpl3vpn-ipv4-unicast:
    +++-rw tunnelSelectorName?    string
augment /bgp:bgp/bgpglobal:bgpgafi-safis/bgpgafi-safi/
  bgpl3vpn-ipv6-unicast:
    +++-rw tunnelSelectorName?    string

L. Zhenbin, et al  [Page 8]
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 tnlmbaseTnlPolicyType {
  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 tnlmbaseSelTnlType {
  type enumeration {
    enum "invalid" {
      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 tnlmbaseTnlPolicyType;
  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')" + "
           + "(./.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
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 tnlmbaseSelTnlType;
  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 tnlmbaseSelTnlType;
  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 tnlmbaseSelTnlType;
  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 tnlmbaseSelTnlType;
  default "invalid";
  description
    "Sequence in which different types of tunnels are selected. If the value is INVALID, no tunnel type has been configured.";
selected. If the value is INVALID, no tunnel type has been configured.

leaf selTnlType5 {
    when "not(../selTnlType1='invalid' or "
        + "+../selTnlType2='invalid' or 
        + "+../selTnlType3='invalid' or 
        + "+../selTnlType4='invalid' or 
        + "+../selTnlType5='invalid')"
    type tnlmbaseSelTnlType;
    default "invalid";
    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='invalid' or "
        + "+../selTnlType2='invalid' or 
        + "+../selTnlType3='invalid' or 
        + "+../selTnlType4='invalid' or 
        + "+../selTnlType5='invalid')"
    type tnlmbaseSelTnlType;
    default "invalid";
    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.

L. Zhenbin, et al

[Page 17]
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).

```yang
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.";
  }
}

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 " + ",to/to\"matchNHopPF\"";
    type "string";
  }
}

L. Zhenbin, et al
leaf aclNameOrNum {
    when "not(../matchType='matchNHopPF' or "
        + "not(../matchType)) or "
        + ".../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-zA-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]*[0-9][0-9])|([0]*[0-9][0-9]))’
            }
    }
} //End of container matchCommunityFilters

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


Normative References

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.
Identification of Overlay Operations, Administration, and Maintenance (OAM)
draft-mirsky-rtgwg-oam-identify-03

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 February 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.
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 Channeled 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. Overlay Network Encapsulations

New overlay network encapsulations analyzed in two groups:

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

3.1. 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.mmbb-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:

O 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 O 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., O
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 O 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.

3.1.1. 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 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.

3.2. 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.
3.3. 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:

- if available, use Sender ID in the overlay domain (example BFIR ID in BIER [RFC8296]);
- use IP/UDP encapsulation of an OAM packet in the overlay (similar to Section 4.3 [RFC8029]).

3.4. 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 3.1. 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 3, BIER allocated the field that may be used for AMM, as discussed in [I-D.ietf-bier-pmm-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.

4. 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:
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.

OAM that follows the overlay header identified as payload type, e.g., by the value of the Next Protocol field.

5. IANA Considerations

This document does not propose any IANA consideration. This section may be removed.

6. 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.

7. Acknowledgment

TBD

8. References

8.1. Normative References


8.2. Informational References

[I-D.fioccola-v6ops-ipv6-alt-mark]

[I-D.fmm-nvo3-pm-alt-mark]


Mirsky, G., Xiao, M., Boutros, S., and D. Black, "OAM for use in GENEVE", draft-mmbb-nvo3-geneve-oam-00 (work in progress), July 2019.


Author’s Address

Greg Mirsky
ZTE Corp.

Email: gregimirsky@gmail.com
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/.
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 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 to 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

7.2. Informative References


Authors' Addresses

Loa Andersson
Bronze Dragon Consulting

Email: loa@pi.nu
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.
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


9.2. Informative References

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
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
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:

- OSE Gateway Reachability services
- 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:

- client
- server
- configuration data
- state data

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

- augment
- data model
- data node

The terminology for describing YANG data models is found in [RFC7950].
Internet-Draft            SDWAN OSE YANG Model                 June 2019

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 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 SD-WAN network. The SD-WAN network controller delivers control-plane/data-plane separation that 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 SD-WAN 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:

- 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.

- 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
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 Orchestration 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:

- Create one or more OSE gateways in the serving domain.
- Create underlying connections between the OSE gateway and other SD-WAN domain gateways, including control plane and data plane.
- 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.
- 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.

- ose-gateway: Gateway name and Gateway ID are specified for each domain.
- tunnel: describes encap-type in the interconnection points, and authentication and encryption are also specified to secure the interconnection between SD-WAN domains.
- ose-interworking-option: MP-BGP based L3VPN Inter-AS Option B with MPLS labels and Inter-AS Option A are defined.
- 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.
- 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:
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.

```
+-------------------+       +------------------+
|                  |   \ Link & Path   |     |    Link&Path     |
|Traffic Classifier+----+     Policy       +------+  Performance     |
|                  |   /     |     SLAs         |
+-------------------+       +------------------+
```

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

---

Wood, et al.

Expires December 31, 2019

[Page 10]
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:

- **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
- **physical-port**: describe the WAN port number
- **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.
- **service-provider**: Specifies the name of provider per enumerated list of providers globally.
- **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:

- **delay**: defines the latency constraint of a specific traffic class.
- **jitter**: defines the jitter constraint of a specific traffic class.
- **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 {
    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
enum gre {
    description
    "The encapsulation option is GRE tunnel per.";
}

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.";
}

leaf 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.";
  }
}
"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.";
}
"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.";
}
"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."
}
}

"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.";
      }
    }
  }
}
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.";
    }
  }
  description
    "When in NOT automatic mode, specify the physical-port, service-type.";
}
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";
}

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:

- /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.

- /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.

- /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.

- /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:

---------------------------------------------------------------------
Registrant Contact: The IESG
XML: N/A; the requested URI is an XML namespace.

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
Prefix: path-svc
Reference: RFC xxxx
Name: ietf-ose-gateway-svc
Prefix: reach-vpn
Reference: RFC xxxx
---------------------------------------------------------------------
11. References

11.1. Normative References


11.2. Informative References


Appendix A. Acknowledges

Authors’ Addresses

Steve Wood (editor)
Cisco Systems
Email: swood1@cisco.com

Bo Wu (editor)
Huawei Technologies, Co., Ltd
Email: lana.wubo@huawei.com

Qin Wu (editor)
Huawei Technologies, Co., Ltd
Email: bill.wu@huawei.com

Conrad Menezes
HPE Aruba
Email: conrad.menezes@hpe.com
A Framework for Automating Service and Network Management with YANG

draft-wu-model-driven-management-virtualization-06

Abstract

Data models for service and network management provide 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.
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:

- 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.

- Secondly, many current service fulfillment might not support real time streaming telemetry capability in high frequency and in high throughput on the current state of networking 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
Internet-Draft  Service and Network Management Automation   October 2019

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:

- 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.

- Techniques for exposing network services [RFC8309] and their 
characteristics.

- Techniques used by service-requirement-derived dynamic resource 
allocation and policy enforcement schemes, so that networks can be 
programmed accordingly.

- 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.

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].

2. Terminology

The following terms are defined in [RFC8309][RFC8199] and are not redefined here:

- Network Operator
- Customer
- Service
- Data Model
- Service Model
- Network Element Module

The document makes use of the following terms:

Network Resource Model: is used by a network operator to allocate a network resource (e.g., tunnel resource, topology resource) for the service or schedule the resource to meet the service requirements captured in a 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 the IETF can be classified into service level, network level, and device level modules. Different service
level modules may rely on the same set of network level or device level modules.

Service level modules usually follow top down approach and are mostly customer-facing modules providing a common model construct for higher level network services (e.g., L3VPN), which can be further mapped to network technology-specific modules at lower layer (e.g., tunnel, routing, QoS, security). For example, the service level can be used to characterise the network service(s) to be ensured between service nodes (ingress/egress) such as the communication scope (pipe, hose, funnel, ...), the directionality, the traffic performance guarantees (one-way delay (OWD), one-way loss, ...), etc.

Network level modules mostly follow a bottom-up approach and 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 traffic-engineering Tunnel modules).

Device (and function) level modules usually follow a bottom-up approach and are mostly technology-specific modules used to realize a service (e.g., BGP, NAT).

Each level maintains a view of the supported YANG modules provided by low-levels (see for example, Appendix A).

Figure 1 illustrates the overall layering model.
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 Fulfillment 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 2.
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. 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.

4.1.2. 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.

4.1.3. 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:

- monitor network communications (i.e., reachability verification and Continuity Check)
- troubleshoot failures (i.e., fault verification and localization)
- 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:

- 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. Operational State Telemetry

<operational> datastore holds the complete operational state of the device including learned, system, default configuration and system state. <operational> datastore can be used as telemetry data source and allows the client subscribe to updates of a YANG datastore.

Based on criteria negotiated as part of a subscription, updates will be pushed to targeted recipients using YANG push mechanism [RFC8641].

4.2.4. Fault Diagnostic

Technology-dependent nodes and remote procedure call (RPC) commands are defined in technology-specific YANG modules which use and extend the base model described in Section 4.1.4.

These RPC commands received in the technology dependent node can be used to trigger technology specific OAM message exchange for fault verification and fault isolation.
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
In reference to Figure 3, 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

In reference to Figure 4, 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:

- Create forwarding loops by mis-configuring the underlying network.
- Leak sensitive information: special care should be considered when translating between the various layers introduced in the document.
- ...

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. Informative References

[I-D.arkko-arch-virtualization]

[I-D.asechoud-netmod-diffserv-model]

[I-D.clacla-netmod-model-catalog]
[I-D.homma-slice-provision-models]

[I-D.ietf-bess-evpn-yang]

[I-D.ietf-bess-l2vpn-yang]

[I-D.ietf-bess-l3vpn-yang]

[I-D.ietf-bfd-yang]

[I-D.ietf-ccamp-alarm-module]

[I-D.ietf-ccamp-flexigrid-media-channel-yang]

[I-D.ietf-ccamp-flexigrid-yang]
Internet-Draft  Service and Network Management Automation   October 2019

[I-D.ietf-ccamp-l1csm-yang]  

[I-D.ietf-ccamp-mw-yang]  

[I-D.ietf-ccamp-otn-topo-yang]  

[I-D.ietf-ccamp-otn-tunnel-model]  

[I-D.ietf-ccamp-wson-tunnel-model]  

[I-D.ietf-dots-data-channel]  

[I-D.ietf-dots-signal-channel]  

[I-D.ietf-idr-bgp-model]  


I-D.ietf-softwire-iftunnel

I-D.ietf-softwire-yang

I-D.ietf-spring-sr-yang

I-D.ietf-supap-generic-policy-data-model

I-D.ietf-teas-actn-vn-yang

I-D.ietf-teas-sf-aware-topo-model

I-D.ietf-teas-te-service-mapping-yang

I-D.ietf-teas-yang-13-te-topo

I-D.ietf-teas-yang-path-computation


Appendix A. Layered YANG Modules Example Overview

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:

- L3SM model [RFC8299] defines the L3VPN service ordered by a customer from a network operator.
- L2SM model [RFC8466] defines the L2VPN service ordered by a customer from a network operator.
- 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 5 depicts a set of Network models such as topology models or tunnel models:

<table>
<thead>
<tr>
<th>Topo YANG modules</th>
<th>Tunnel YANG modules</th>
</tr>
</thead>
<tbody>
<tr>
<td>+------------+----+ +------+-+----+</td>
<td></td>
</tr>
<tr>
<td>Network Top</td>
<td>Other</td>
</tr>
<tr>
<td>Model</td>
<td>Tunnel</td>
</tr>
<tr>
<td>+----+----+</td>
<td>+----+----+</td>
</tr>
<tr>
<td>Svc Topo</td>
<td>L2 Topo</td>
</tr>
<tr>
<td>+----+----+</td>
<td>+----+----+</td>
</tr>
<tr>
<td>L2 Topo</td>
<td>TE Topo</td>
</tr>
<tr>
<td>+----+----+</td>
<td>+----+----+</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 5: Sample Resource Facing Network Models

Topology YANG module Examples:

- **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.

- **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.

- **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.
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:

- Tunnel identities [I-D.ietf-softwire-iftunnel] to ease manipulating extensions to specific tunnels.

- TE Tunnel Model

  [I.D-ietf-teas-yang-te] defines a YANG module for the configuration and management of TE interfaces, tunnels and LSPs.

- 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.

- 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.

- 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:

- 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].

- 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.

- **Generic Policy Model**

  The Simplified Use of Policy Abstractions (SUPA) policy-based management framework [RFC8328] defines base YANG modules [I-D.ietf-supap-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 6) 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.
A.3.1. Model Composition

- 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.

- 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.

- 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-rtwg-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-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
Huawei
101 Software Avenue, Yuhua District
Nanjing, Jiangsu 210012
China

Email: bill.wu@huawei.com

Mohamed Boucadair
Orange
Rennes 35000
France

Email: mohamed.boucadair@orange.com

Christian
Orange
Rennes 35000
France

Email: christian.jacquenet@orange.com

Luis Miguel Contreras Murillo
Telefonica

Email: luismiguel.contrerasmurillo@telefonica.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

Weiqiang Cheng  
China Mobile  
Email: chengweiqiang@chinamobile.com

Liang Geng  
China Mobile  
Email: gengliang@chinamobile.com

Young Lee  
Futurewei  
Email: younglee.tx@gmail.com