

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 3, 2019

Y. Gu
S. Zhuang
Z. Li
Huawei
July 02, 2018

Network Monitoring Protocol (NMP)
draft-gu-network-mornitoring-protol-00

Abstract

To enable automated network OAM (Operations, administration and management), the availability of network protocol running status information is a fundamental step. In this document, a network monitoring protocol (NMP) is proposed to provision the information related to running status of IGP (Interior Gateway Protocol) and other control protocols. It can facilitate the network troubleshooting of control protocols in a network domain. Typical network issues are illustrated as the usecases of NMP for ISIS to showcase the necessity of NMP. Then the operations and the message formats of NMP for ISIS are defined. In this document ISIS is used as the illustration protocol, and the case of OSPF and other control protocols will be included in the future version.

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 3, 2019.

Copyright Notice

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 (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
1.1. Motivation	2
1.2. Overview	3
2. Terminology	4
3. Use Cases	4
3.1. ISIS Adjacency Issues	4
3.2. Forwarding Path Disconnection	5
3.3. ISIS LSP Synchronization Failure	5
4. Extensions of NMP for ISIS	6
4.1. Message Types	6
4.2. Message Format	7
4.2.1. Common Header	7
4.2.2. Per Peer Header	7
4.2.3. Initiation Message	8
4.2.4. Peer Status Change Notification	9
4.2.5. Statistic Report Message	10
4.2.6. ISIS PDU Monitoring Message	12
4.2.7. Termination Message	12
5. IANA	13
6. Contributors	13
7. Acknowledgments	13
8. References	13
Authors' Addresses	15

1. Introduction

1.1. Motivation

The requirement for better network OAM approaches has been greatly driven by the network evolution. Network OAM provides visibility to the network health conditions, and is beneficial for faster network

troubleshooting and self-healing, network OpEx (operating expenditure) reduction, and network optimization. Network OAM statistics show that a relatively large part of the network issues are caused by the disfunction of various routing protocols and MPLS signalings.

The general troubleshooting logic nowadays is to log in a faulty router, physically or through Telnet, and by using CLI to display related information/logs for fault source localization and further analysis. There are several concerns with the conventional troubleshooting:

1. It requires rich OAM experience for the OAM operator to know what information to check on the device, and the operation is complex;
2. In a multi-vendor network, it requires the understanding and familiarity of vendor specific operations and configurations;
3. Locating the fault source device could be non-trivial work, and is often realized through network-wide device-by-device check, which is both time-consuming and labor-consuming; and finally,
4. The acquisition of troubleshooting data can be difficult under some cases, e.g., when auto recovery is used.

Alternatively, the idea of collecting information from devices and exporting to the centralized controller/server for further analysis is also used to gain more insight on the management plane information for OAM purposes. For example, SNMP (Simple Network Management Protocol) [RFC1157], NETCONF (Network Configuration Protocol) [RFC6241], gNMI/gRPC [I-D.openconfig-rtgwg-gnmi-spec], etc. are used for the purpose. However, the approaches are mainly used for data SET/GET of the management plane which are insufficient for the troubleshooting of control plane issues.

BGP monitoring protocol (BMP) [RFC7854] has been proposed to monitor BGP routes and peer status which provides the control plane information and thus more insight for troubleshooting. This document extends BMP to collect information of other control protocols for monitoring to facilitate the trouble shooting of control plane issues which call as Network Monitoring Protocols (NMP).

1.2. Overview

Like BMP, an NMP session is established between each monitored router (NMP client) and the NMP monitoring station (NMP server) through TCP connection. Information are collected directly from each monitored

router and reported to the NMP server. The NMP message can be both periodic and event-triggered, depending on the message type.

ISIS [RFC1195], as one of the most commonly adopted network layer protocols, builds the fundamental network connectivity of an autonomous system (AS). The disfunction of ISIS, e.g., ISIS neighbor down, route flapping, MTU mismatch, and so on, could lead to network-wide instability and service interruption. Thus, it is critical to keep track of the health condition of ISIS, and the availability of information, related to ISIS running status, is the fundamental requirement. In this document, typical network issues are illustrated as the use cases of NMP for ISIS to showcase the necessity of NMP. Then the operations and the message formats of NMP for ISIS are defined. In this document ISIS is used as the illustration protocol, and the case of OSPF and other control protocols will be included in the future version.

2. Terminology

IGP: Interior Gateway Protocol

NMP: Network Monitoring Protocol

IMP: Network Monitoring Protocol for IGP

3. Use Cases

We have identified several typical network issues due to ISIS disfunction that are currently difficult to detect or localize. The usage of NMP is not limited to the solve the following listed issues.

3.1. ISIS Adjacency Issues

ISIS adjacency issues are identified as top network issues and may take hours to localize. The adjacency issues can be classified into two situations:

1. An existing established adjacency goes down;
2. An adjacency fails to be established.

In Case 1, the adjacency down can be caused by factors such as circuit down, hold timer expiration, device memory low, user configuration change, and so on. Case 2 can be caused by mismatch link MTU, mismatch authentication, mismatch area ID, system ID conflict, and so on. Typically, such adjacency failure events are logged/recorded in the device, but currently there is no real-time report/alarm of such issue. The conventional troubleshooting process

for adjacency issue is to find the faulty devices and then log in to check the logs or the Hello statistics for further analysis.

Using NMP, the ISIS adjacency status: up, down and initial, is reported to the NMP server in real time, together with the possible recorded reasons. Then the NMP server can solve such issue in about minutes. For example, for an adjacency set up failure due to different authentications, the NMP server can recognize the difference by comparing the Hello PDUs collected from both devices.

3.2. Forwarding Path Disconnection

Mismatched MTU values for devices along a certain path can lead to packet forwarding failure while the control plane is working properly. The failure may not be detected by Ping, but the forwarding plane appears disconnected for certain size of data packets. It can be quite common since vendors have different understanding and configuration of MTU. There are methods proposed to discover the path MTU. For example, router's link MTU is conveyed in the MPLS LDP/RSVP-TE path set up signaling, and the path MTU is decided at the ingress or egress node[RFC3988] [RFC3209]. For IPv4 packets, by setting the DF flag bit of the outgoing packet, any device along the path with smaller MTU will drop the packet, and send back an ICMP Fragmentation Needed message containing its MTU, allowing the source to reduce the MTU. The process is repeated until the MTU is small enough to traverse the entire path without fragmentation[RFC1191]. Apparently, such method is too time-consuming.

Using NMP, each device can report its link MTU to the monitoring station directly. The mismatch can be recognized at the NMP server in seconds.

3.3. ISIS LSP Synchronization Failure

It happens that two ISIS neighbors fail to learn the LSPs sent from each other in the following two cases: in Case 1, the LSP fails to be received, and in Case 2, the LSP is received but the LSP information shown in the receiver's LSDB is not the same as the one sent from the transmitter (e.g., one or more prefixes missing, the LSP sequence number modified). Case 1 can be caused by link failure, similar to the adjacency down issue. In Case 2, the received LSP can be processed incorrectly due to hardware/software bugs. In fact, the LSDB synchronization issue is usually hard to localize once happens.

Using NMP, the NMP server can detect the failure by comparing the sent/received LSP statistics from the two neighbors. In the case that the received LSPs are improperly processed within the device,

the NMP monitoring station can recognize the LSP synchronization failure by comparing the LSPs sent out from the two neighbors.

4. Extensions of NMP for ISIS

4.1. Message Types

The variety of ISIS troubleshooting use cases requires a systematic information report of NMP, so that the NMP server or any third party analyzer could efficiently utilize the reported messages to localize and recover various network issues. We define NMP messages for ISIS uses the following types:

- o **Initiation Message:** A message used for the monitored device to inform the NMP monitoring station of its capabilities, vendor, software version and so on. For example, the link MTU can be included within the message. The initiation message is sent once the TCP connection between the monitoring station and monitored router is set up. During the monitoring session, any change of the initiation message could trigger an Initiation Message update.
- o **Peer Status Change Notification Message:** A message used to inform the monitoring station of the adjacency status change of the monitored device, i.e., from up to down, from down/initiation to up, with possible alarms/logs recorded in the device. This message notifies the NMP server of the ongoing ISIS adjacency change event and possible reasons. If no reason is provided or the provided reason is not specific enough, the NMP server can further analyze the ISIS PDU or the ISIS statistics.
- o **Statistic Report Message:** A message used to report the statistics of the ongoing ISIS process at the monitored device. For example, abnormal LSP count of the monitored device can be a sign of route flapping. This message can be sent periodically or event triggered. If sent periodically, the frequency can be configured by the operator depending on the monitoring requirement. If it's event triggered, it could be triggered by a counter/timer exceeding the threshold.
- o **ISIS PDU Monitoring Message:** A message used to update the NMP server of any PDU sent from and received at the monitored device. For example, the Hello PDUs collected from two neighbors can be used for analyzing the adjacency set up failure issue. The LSPs collected from two neighbors can be analyzed for the LSP synchronization issue.

- o Termination Message: A message for the monitored router to inform the monitoring station of why it is closing the NMP session. This message is sent when the monitoring session is to be closed.

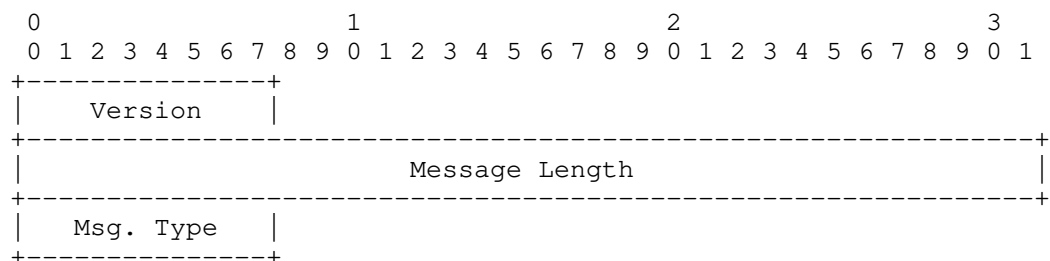
4.2. Message Format

4.2.1. Common Header

The common header is encapsulated in all NMP messages. It includes the Version, Message Length and Message Type fields.

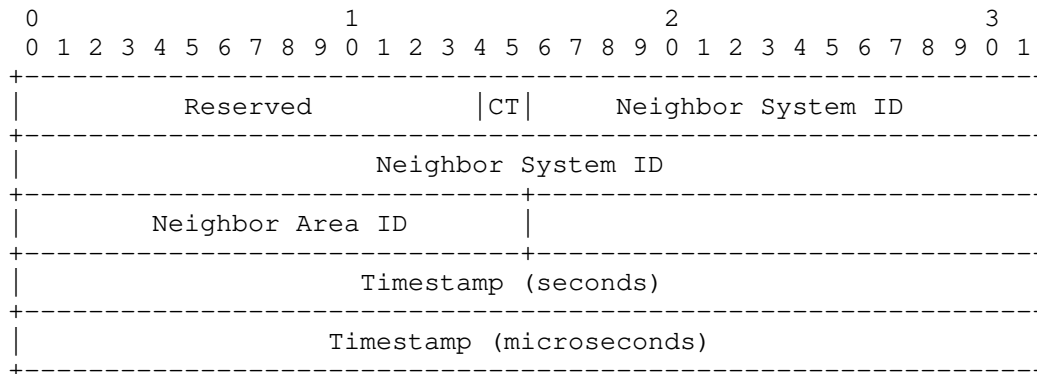
- o Version (1 byte): Indicates the NMP version and is set to '1' for all messages.
- o Message Length (4 bytes): Length of the message in bytes (including headers, data, and encapsulated messages, if any).
- o Message Type (1 byte): This indicates the type of the NMP message, which are listed as follows.

- * Type = 0: Initiation
- * Type = 1: Peer Status Change Notification
- * Type = 2: Statistic Report
- * Type = 3: ISIS PDU Monitoring
- * Type = 4: Termination Message



4.2.2. Per Peer Header

Except the Initiation and Termination Message, all the rest messages are per adjacency based. Thus, a per peer header is defined as follows.

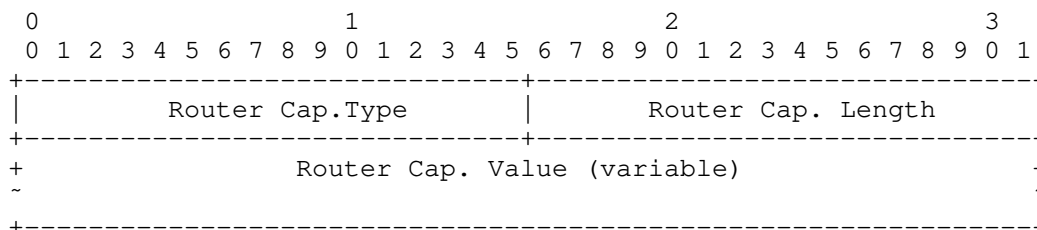


- Peer Flag (2 bytes): The Circuit Type (2 bits) flag specifies if the router is an L1(01), L2(10), or L1/L2(11). If both bits are zeroes (00), the Per Peer Header is ignored. This configuration is used when the statistic is not per-peer based, e.g., when reporting the number of adjacencies.
- Neighbor System ID (6 bytes): identifies the system ID of the remote router.
- Neighbor Area ID (2 bytes): identifies the area ID of the remote router.
- Timestamp (4 bytes): records the time when the message is sent/received, expressed in seconds and microseconds since midnight (zero hour), January 1, 1970 (UTC).

4.2.3. Initiation Message

The Initiation Message indicates the monitored router's capabilities, vendor, software version and so on. It consists of the Common Header and the Router Capability TLV. The Common Header can be followed by multiple Router Capability TLVs.

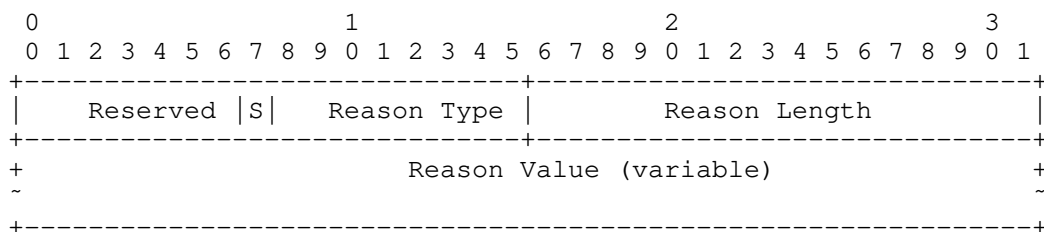
The Router Capability TLV is defined as follows.



- o Router Capability Type: provides the type of the router capability information. Currently defined types are:
 - * Type = 0: sysDescr. The corresponding Router Capability Value field should contain an ASCII string whose value MUST be set to be equal to the value of the sysDescr MIB-II [RFC1213] object.
 - * Type = 1: sysName. The corresponding Router Capability Value field should contain an ASCII string whose value MUST be set to be equal to the value of the sysName MIB-II [RFC1213] object.
 - * Type = 2: Local System ID. The corresponding Router Capability Value field should indicate the router's System ID
 - * Type = 3: Link MTU. The corresponding Router Capability Value field should indicate the router's link MTU.
 - * Type = 4: String. The corresponding Router Capability Value field contains a free-form UTF-8 string whose length is given by the Information Length field.

4.2.4. Peer Status Change Notification

The Peer Status Change Notification Message indicates an ISIS adjacency status change: from up to down or from initiation/down to up. It consists of the Common Header, Per Peer Header and the Reason TLV. The Notification is triggered whenever the status changes. The Reason TLV is optional, and is defined as follows. More Reason types can be defined if necessary.



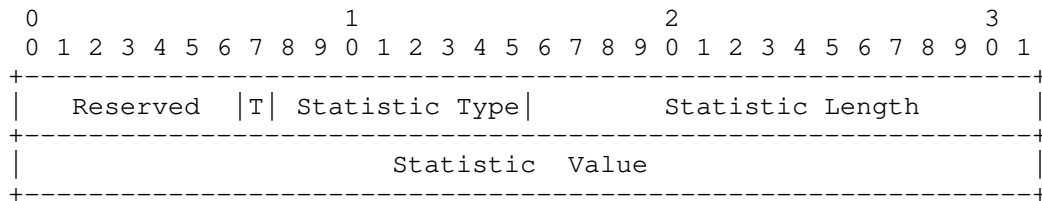
- o Reason Flags (1 byte): The S flag (1 bit) indicates if the Peer status is from up to down (set to 0) or from down/initial to up (set to 1). The rest bits of the Flag field are reserved. When the S flag is set to 1, the Reason Type should be set to all zeroes (i.e., Type 0), the Reason Length fields should be set to all zeroes, and the Reason Value field should be set empty.
- o Reason Type (1 byte): indicates the possible reason that caused the peer status change. Currently defined types are:

- * Type = 0: Adjacency Up. This type indicates the establishment of an adjacency. For this reason type, the S flag MUST be set to 1, indicating it's a peer-up event. There's no further reason to be provided. The reason Length field should be set to all zeroes, and the Reason Value field should be set empty.
 - * Type = 1: Circuit Down. For this data type, the S flag MUST be set to 0, indicating it's a peer-down event. The length field is set to all zeroes, and the value field is set empty.
 - * Type = 2: Memory Low. For this data type, the S flag MUST be set to 0, indicating it's a peer-down event. The length field is set to all zeroes, and the value field is set empty.
 - * Type = 3: Hold timer expired. For this data type, the S flag MUST be set to 0, indicating it's a peer-down event. The length field is set to all zeroes, and the value field is set empty.
 - * Type = 4: String. For this data type, the S flag MUST be set to 0, indicating it's a peer-down event. The corresponding Reason Value field indicates the reason specified by the monitored router in a free-form UTF-8 string whose length is given by the Reason Length field.
- o Reason Length (2 bytes): indicates the length of the Reason Value field.
 - o Reason Value (variable): includes the possible reason why the Adjacency is down.

4.2.5. Statistic Report Message

The Statistic Report Message reports the statistics of the parameters that are of interest to the operator. The message consists of the NMP Common Header, the Per Adjacency Header and the Statistic TLV. The message include both per-peer based statistics and non per-peer based statistics. For example, the received/sent LSP counts are per-peer based statistics, and the local LSP change times count and the number of established adjacencies are non per-peer based statistics. For the non per-peer based statistics, the CT Flag (2 bits) in the Per Peer Header MUST be set to 00. Upon receiving any message with CT flag set to 00, the Per Peer Header should be ignored (the total length of the Per Peer Header is 18 bytes as defined in Section 3.2.2, and the message reading/analysis should resume from the Statistic TLV part.

The Statistic TLV is defined as follows.



- o **Statistic Flags (1 byte):** provides information for the reported statistics.
 - * **T flag (1 bit):** indicates if the statistic is for the received-from direction (set to 1) or sent-to direction the neighbor (set to 0)
- o **Statistic Type (1 byte):** specifies the statistic type of the counter. Currently defined types are:
 - * **Type = 0:** Hello PDU count. The T flag indicates if it's a sent or received Hello PDU. It is a per-peer based statistic type, and the CT flag in the Per Peer Header MUST NOT be set to 00.
 - * **Type = 1:** Incorrect Hello PDU received count. For this type, the T flag MUST be set to 1. It is a per-peer based statistic type, and the CT flag in the Per Peer Header MUST NOT be set to 00.
 - * **Type = 2:** LSP count. The T flag indicates if it's a sent or received LSP. It is a per-peer based statistic type, and the CT flag in the Per Peer Header MUST NOT be set to 00.
 - * **Type = 3:** Incorrect LSP received count. For this type, the T flag MUST be set to 1. It is a per-peer based statistic type, and the CT flag in the Per Peer Header MUST NOT be set to 00.
 - * **Type = 4:** Retransmitted LSP count. For this type, the T flag MUST be set to 0. It is a per-peer based statistic type, and the CT flag in the Per Peer Header MUST NOT be set to 00.
 - * **Type = 5:** CSNP count. The T flag indicates if it's a sent or received CSNP. It is a per-peer based statistic type, and the CT flag in the Per Peer Header MUST NOT be set to 00.
 - * **Type = 6:** PSNP count. The T flag indicates if it's a sent or received PSNP. It is a per-peer based statistic type, and the CT flag in the Per Peer Header MUST NOT be set to 00.

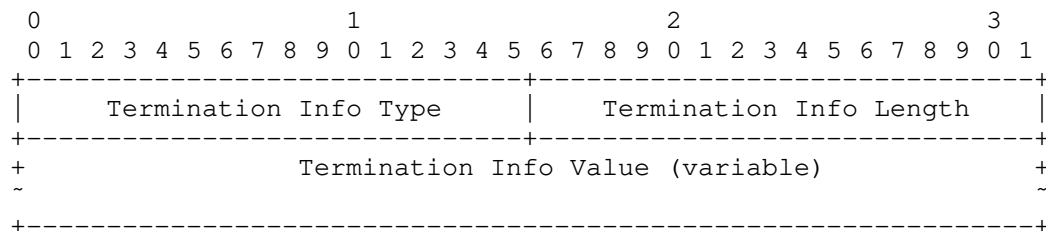
- * Type = 7: Number of established adjacencies. It's a non per-peer based statistic type, and thus for the monitoring station to recognize this type, the CT flag in the Per Peer Header MUST be set to 00.
- * Type = 8: LSP change time count. It's a non per-peer based statistic type, and thus for the monitoring station to recognize this type, the CT flag in the Per Peer Header MUST be set to 00.
- o Statistic Length (2 bytes): indicates the length of the Statistic Value field.
- o Statistic Value (4 bytes): specifies the counter value, which is a non-negative integer.

4.2.6. ISIS PDU Monitoring Message

The ISIS PDU Monitoring Message is used to update the monitoring station of any PDU sent from and received at the monitored device per neighbor. Following the Common Header and the Per Peer Header is the ISIS PDU. To tell whether it's a sent or received PDU, the monitoring station can analyze the source and destination addresses in the reported PDUs.

4.2.7. Termination Message

The Termination Message is sent when the NMP session is to be closed, and is used to indicate the termination reason to the monitoring station. The TCP session between the monitored router and the monitoring station should be terminated upon receiving this message. It consists of the Common Header and the Termination Info TLVs, defined as follows.



- o Termination Info Type (2 bytes): Provides the termination reason type. Currently defined types are:
 - * Type = 0: Unknown. This reason type specifies that the NMP session is closed for an unknown or unspecified reason. For

this data type, the length field is filled with all zeroes, and the value field is set empty.

- * Type = 1: Memory Low. This reason indicates that the monitored router lacks resources for the NMP session. For this data type, the length field is filled with all zeroes, and the value field is set empty.
- * Type = 2: Administratively Closed. This reason specifies that the session is closed due to administrative reasons. The corresponding Termination Info Value field may include more details about the reason expressed in a free-form UTF-8 string whose length is given by the Termination Info Length field.
- * Type = 3: String. The corresponding Termination Info Value field may include details about the reason expressed in a free-form UTF-8 string whose length is given by the Termination Info Length field.

Termination Info Length (2 bytes): indicates the length of the Termination Info Reason Value field.

- o Termination Info Value (variable): includes more detailed reason for the session termination.

5. IANA

TBD

6. Contributors

TBD

7. Acknowledgments

TBD

8. References

[I-D.ietf-netconf-yang-push]

Clemm, A., Voit, E., Prieto, A., Tripathy, A., Nilsen-Nygaard, E., Bierman, A., and B. Lengyel, "YANG Datastore Subscription", draft-ietf-netconf-yang-push-17 (work in progress), July 2018.

- [I-D.openconfig-rtgwg-gnmi-spec]
Shakir, R., Shaikh, A., Borman, P., Hines, M., Lebsack, C., and C. Morrow, "gRPC Network Management Interface (gNMI)", draft-openconfig-rtgwg-gnmi-spec-01 (work in progress), March 2018.
- [RFC1157] Case, J., Fedor, M., Schoffstall, M., and J. Davin, "Simple Network Management Protocol (SNMP)", RFC 1157, DOI 10.17487/RFC1157, May 1990, <<https://www.rfc-editor.org/info/rfc1157>>.
- [RFC1191] Mogul, J. and S. Deering, "Path MTU discovery", RFC 1191, DOI 10.17487/RFC1191, November 1990, <<https://www.rfc-editor.org/info/rfc1191>>.
- [RFC1195] Callon, R., "Use of OSI IS-IS for routing in TCP/IP and dual environments", RFC 1195, DOI 10.17487/RFC1195, December 1990, <<https://www.rfc-editor.org/info/rfc1195>>.
- [RFC1213] McCloghrie, K. and M. Rose, "Management Information Base for Network Management of TCP/IP-based internets: MIB-II", STD 17, RFC 1213, DOI 10.17487/RFC1213, March 1991, <<https://www.rfc-editor.org/info/rfc1213>>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC3209] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V., and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, DOI 10.17487/RFC3209, December 2001, <<https://www.rfc-editor.org/info/rfc3209>>.
- [RFC3988] Black, B. and K. Kompella, "Maximum Transmission Unit Signalling Extensions for the Label Distribution Protocol", RFC 3988, DOI 10.17487/RFC3988, January 2005, <<https://www.rfc-editor.org/info/rfc3988>>.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <<https://www.rfc-editor.org/info/rfc6241>>.

- [RFC7752] Gredler, H., Ed., Medved, J., Previdi, S., Farrel, A., and S. Ray, "North-Bound Distribution of Link-State and Traffic Engineering (TE) Information Using BGP", RFC 7752, DOI 10.17487/RFC7752, March 2016, <<https://www.rfc-editor.org/info/rfc7752>>.
- [RFC7854] Scudder, J., Ed., Fernando, R., and S. Stuart, "BGP Monitoring Protocol (BMP)", RFC 7854, DOI 10.17487/RFC7854, June 2016, <<https://www.rfc-editor.org/info/rfc7854>>.

Authors' Addresses

Yunan Gu
Huawei
156 Beiqing Road
Beijing, 100095
P.R. China

Email: guyunan@huawei.com

Shunwan Zhuang
Huawei
156 Beiqing Road
Beijing, 100095
P.R. China

Email: zhuangshunwan@huawei.com

Zhenbin Li
Huawei
156 Beiqing Road
Beijing, 100095
P.R. China

Email: lizhenbin@huawei.com

Network Working Group
Internet-Draft
Intended status: Informational
Expires: December 30, 2018

C. Li
C. Xie
China Telecom
R. Kumar
R. Lohiya
Juniper Networks
G. Fioccola
Telecom Italia
W. Xu
W. Liu
Huawei Technologies
D. Ma
ZDNS
J. Bi
Tsinghua University
June 28, 2018

Coordinated Address Space Management architecture
draft-li-opsawg-address-pool-management-arch-01

Abstract

IP addresses work as a basic element for providing broadband network services. However, the increase in number, diversity and complexity of modern network devices and services creates unprecedented challenges for the currently prevailing approach of manual IP address management. Manually maintaining IP addresses could always be sub-optimal for IP resource utilization. Besides, it requires heavy human effort from network operators. To achieve high utilization and flexible scheduling of IP network addresses, it is necessary to automate the address scheduling process. This document describes an architecture for the IP address space management. It includes architectural concepts and components used in the CASM (Coordinated Address Space Management), with a focus on those interfaces to be standardized in the IETF.

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 30, 2018.

Copyright Notice

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 (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
2. Terminology	4
3. CASM Reference architecture	4
4. The overall procedure of CASM	7
5. CASM Interface and operation	8
5.1. CASM App-facing Interface	8
5.1.1. Functional requirements	8
5.1.2. Interface modeling requirements	9
5.2. CASM device-facing Interface	9
5.2.1. Functional requirements	10
5.2.2. Interface modeling requirements/Initial Address Pool Configuration	10
5.2.3. Interface modeling requirements/Address Pool Status Report	12
5.2.4. Interface modeling requirements/Address Pool Status Query	13
5.2.5. Interface modeling requirements/Address Exhaustion	13
5.2.6. Interface modeling requirements / Address Pool Release	14
6. Services SDN Management Use Cases	15
7. Security Considerations	16
8. Acknowledgements	16
9. References	16
9.1. Normative References	16

9.2. Informative References	17
Authors' Addresses	17

1. Introduction

The address space management is an integral part of any network management solution. However, the increase in number, diversity and complexity of modern network devices and services creates unprecedented challenges for the currently prevailing approach of manual IP address management. Manually maintaining IP addresses could always be sub-optimal for IP resource utilization. Besides, it requires heavy human effort from network operators.

Another factor which drive this work is that tThe network architectures are rapidly changing with the migration toward private and public clouds. At the same time, application architectures are also evolving with a shift toward micro-services and multi-tiered approach.

There is a pressing need to define a new address management system which can meet these diverse set of requirements. To achieve high utilization and flexible scheduling of IP network addresses, Such a system should be capable of automating the address scheduling process. Such a system must be built with well-defined interfaces so users can easily migrate from one vendor to another without rewriting their network management systems.

This document defines a reference architecture that should become the basis to develop a new address management system. This system is called Coodinated Address Space Management (CSAM) system.

A series of use cases are defined in "Use Case Draft". For example, Broadband Network Gateway (BNG), which manages a routable IP address on behalf of each subscriber, should be configured with the IP address pools allocated to subscribers. However, currently operators are facing with the address shortage problem, the remaining IPv4 address pools are usually quite scattered, no more than /24 per address pool in many cases. Therefore, it is complicated to manually configure the address pools on lots of Broadband Network Gateway (BNG) for operators. For large scale Metro Area Network (MAN), the number of BNGs can be up to over one hundred. Manual configuration on all the BNGs statically will not only greatly increase the workload, but also decrease the utilization efficiency of the address pools when the number of subscribers changes over time in the future.

Above is one example of use case, there are other devices which may need to configure address pools as well. In this document, we propose a general mechanism to manage the address pools coordinately,

which can be used in multiple use cases. With this approach, operators do not need to configure the address pools one by one manually and it also helps to use the address pools more efficiently.

2. Terminology

The following terms are used in this document:

CASM: Coordinated Address Space Management, a newly-defined general architecture which can automate IP address management for wide-variety of use cases

IPAM: IP Address Management, a means of planning, tracking, and managing the Internet Protocol address space used in a network

DA: A device agent within the device, which contacts with CASM Coordinator to manipulate address pool

CASM Coordinator: A management system which has a database manage the overall address pools and allocate address pools to devices.

3. CASM Reference architecture

The figure below shows the reference architecture for CASM. This figure covers the various possible scenarios that can exist in future network.

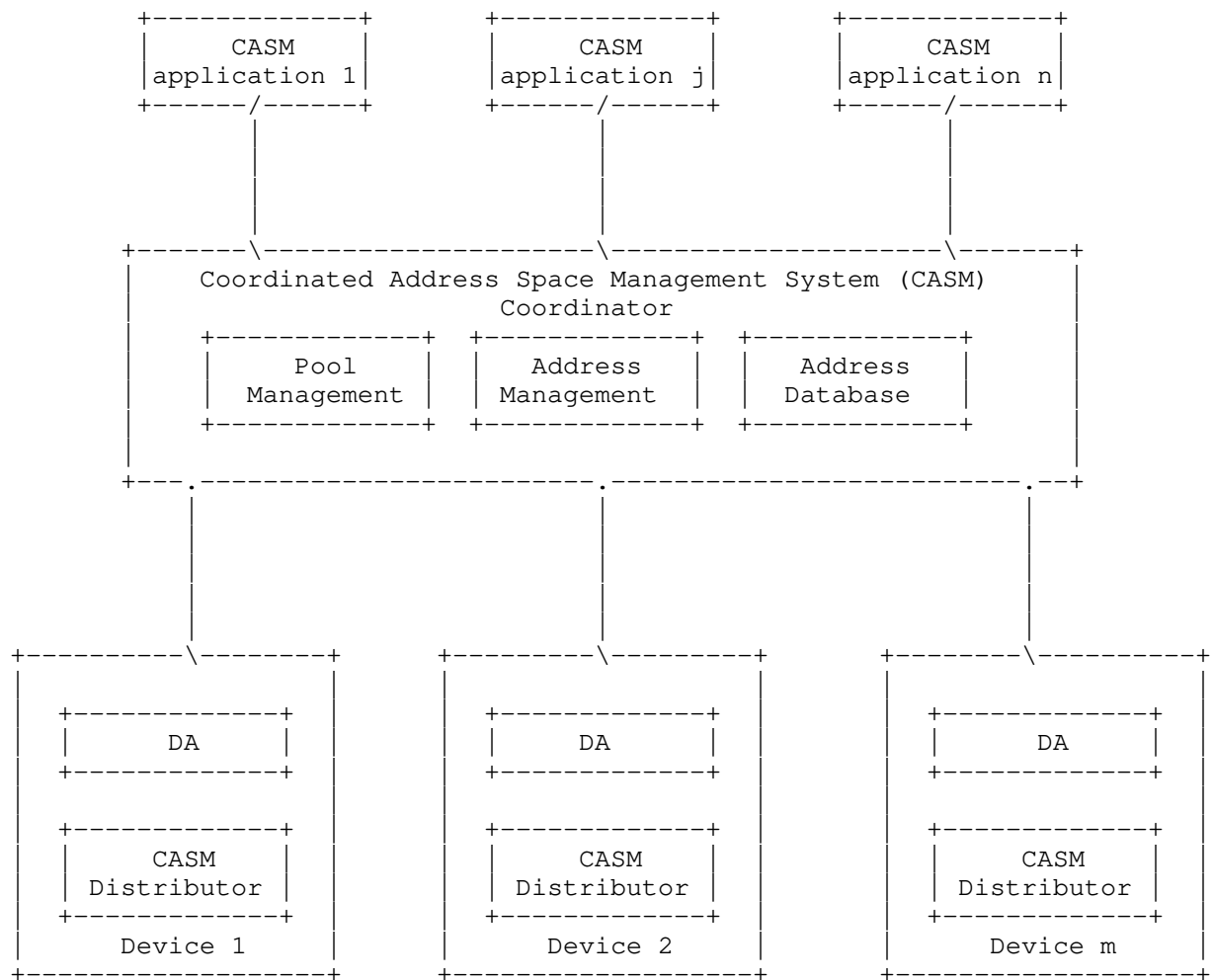


Figure 1: CASM reference architecture

Each component of CASM is introduced as below,

1) CASM Application

The CASM Application is a functional entity which usually has the requirements of centralized address management to realize its specific upper-layer functions. In order to achieve this goal, it needs to manage, operate and maintain the CASM Coordinator. For example, an operator or external user can manage the address pool in

the CASM Coordinator, as well as access log, address allocation records, etc.

2) CASM Coordinator

The CASM Coordinator is a coordinated address management coordinator for the CASM Application to maintain overall address pools, addresses, address properties, etc. It maintains an address database including the overall address pools (OAP) and the address pool status (APS). CASM Applications can maintain their remaining address pools in the OAP. They can also reserve some address pools for special purposes. The address pool status is to reflect the current usage of address pools for different devices. The CASM Coordinator also has the capability to maintain the address pools to different devices dynamically.

3) CASM Device

A CASM Device is responsible for distributing or allocating addresses from local address pools received from the CASM Coordinator. CASM has two components in devices. The first one is Device Agent (DA), which resides in a CASM Device through which the device can contact with the CASM Coordinator. On behalf of the device, the agent initiates the address pool allocation requests, passes the address pools to local instances, detect the availability of address pools or report the status of local address pool usage and update the address pool requests, etc. For some devices, e.g. IPv6 transition and VPN, additional routing modules are needed to update the routing table accordingly.

The CASM Distributor is another component in a CASM device. The DHCP server is a typical distributor that can assign IP addresses to client hosts, and the DHCP protocol is usually used for this task. The address assignment procedure between the CASM Distributor and the client host is out of the scope of this document.

The device determines whether the usage status of the IP address pool resource within the device satisfies the condition. When the IP address pool resource in the device is insufficient or excessive, the device will obtain IP address pool resource request, and sends the request to the CASM Coordinator. The device receives a resource response with IP address pools allocated for it, then it use these address pools to assign IP addresses to end users. Typical CASM Devices include BNGs, BRASes, CGNs, DHCP Servers, NATs, IPv6 Transitions, DNS Servers, etc.

The form of devices is diverse, it can be physical or virtual, and it can be box-integrated with a control plane and a user plane, or a

separated control plane remote from the box, where one or more devices share the centralized control plane. In the latter case, the control plane will manage multiple user plane devices. A number of devices that are subordinate to the control plane will jointly share the address pools to make address utilization much higher.

4. The overall procedure of CASM

1. Operators configure remaining address pools centrally in the CASM Coordinator. There are multiple address pools that can be configured. The CASM Coordinator server then divides the address pools into addressing units (AUs) which would be allocated to device agents by default.
2. The agent will initiate an AddressPool request to the CASM Coordinator. It can carry its desired size of address pool with the request, or just use a default value. The address pool size in the request is only used as a hint. The actual size of the address pool is totally determined by the CASM Coordinator. It would also carry the DA's identification and the type of the address pool.
3. The CASM Coordinator looks up remaining address pools in its local database, and then allocates a set of address pools to the DA. Each address pool has a lifetime.
4. The DA receives the AddressPool reply and uses it for its purpose.
5. If the lifetime of the address pool is going to expire, the DA should issue an AddressPoolRenew request to extend it, including IPv4, IPv6, port numbers, etc.
6. The AddressPoolReport module keeps monitoring and reports the usage of all current address pools for each transition mechanism. If it is running out of address pools, it can renew the AddressPoolRequest for a newly allocated one. It can also release and recycle an existing address pool if that address pool has not been used for a specific and configurable time.
7. When the connection of the CASM Coordinator is lost or it needs the status information of certain applications, it may pre-actively query the DA for its status information.

Currently, the CASM system focuses on the coordination of IP address resources. This Solution should be extended to handle containers, VLAN assignments, etc. These are subject for future work.

5. CASM Interface and operation

5.1. CASM App-facing Interface

The CASM architecture consists of three major distinct entities: CASM Application, CASM Coordinator and network device with a device Agent (DA). In order to provide address space and pools resource that CASM Coordinator can centrally maintain, there is an interface between CASM Applications and CASM Coordinator. The CASM Application can manage the address space and pool in the CASM Coordinator, and the get address allocation records, logs from CASM Coordinator.

5.1.1. Functional requirements

The CASM should support following functionality for it to be adopted for wide variety of use cases.

1. Address pools requirements

A CASM system should allow ability to manage different kind of address pools. The following pools should be considered for implementation; this is not mandatory or exhaustive by any means but given here as most commonly used in networks. The CASM system should allow user-defined pools with any address objects.

Unicast address pool:

- o Private IPv4 addresses
- o Public IPv4 addresses
- o IPv6 addresses
- o MAC Addresses

Multicast address pool:

- o IPv4 address
- o IPv6 address

2. Pool management requirements

There should be a rich set of functionality as defined in this section for operation of a given pool.

Address management:

- o Address allocation either as single or block
- o Address reservation
- o Allocation logic such as mapping schemes or algorithm per pool
- o

General management:

- o Pool initializing, resizing, threshold markings for resource monitoring
- o Pool attributes such as used to automatically create DNS record
- o Pool priority for searching across different pools
- o Pool fragmentation rules, such as how pool can be sub-divided
- o Pool lease rules for allocation requests

5.1.2. Interface modeling requirements

There are three broad categories for CASM interface definition:

Pool management interface: Interface to external user or applications such as SDN controller to manage addresses

Log interface: Interface to access log and records such as DHCP, DNS,
NAT Integration interface: Interface to address services such as
DHCP, DNS, NAT

5.2. CASM device-facing Interface

In order to provide address pool manipulations between CASM Coordinator and device, the CASM architecture calls for well-defined protocols for interfacing between them. Protocol such as radius can be used to compatible with legacy network equipment. And in more modern network system, network device acts as NETCONF/RESTCONF server side, device like CASM Coordinator act as client side. The network device sends address pool request message carrying the requested resource information to the CASM Coordinator, the CASM Coordinator send response message to the network device, where the response message includes address pool resource information allocated to the network device, and network device receives the response message and retrieve the allocated address pool resource information carried in the response message.

5.2.1. Functional requirements

In order to build a complete address management system, it is important that CASM should be able to integrate with other address services. This will provide a complete solution to network operators without requiring any manual or proprietary workflows.

DHCP server:

- o Interface to initialize address pools on DHCP server
- o Notification interface whenever an address lease is modified
- o Interface to access address lease records from DHCP server
- o Ability to store lease records and play back to DHCP server on reboot

DNS server:

- o Interface to create DNS records on DNS server based on DHCP server events

NAT device:

- o Interface to initialize NAT pools
- o Interface to access NAT records from NAT device
- o Ability to store NAT records and play back to NAT device on reboot

5.2.2. Interface modeling requirements/Initial Address Pool Configuration

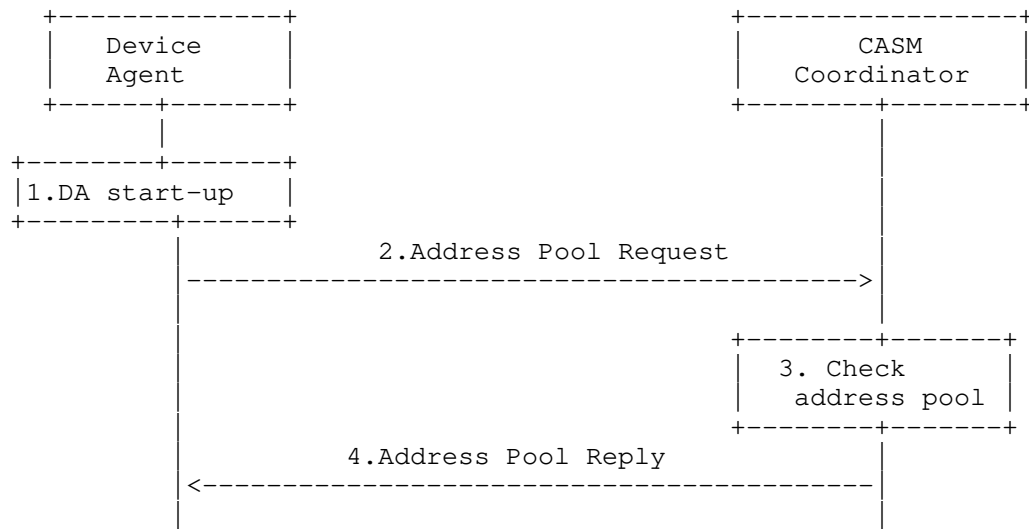


Figure 2: Initial Address Pool Configuration

As shown in Figure 2, the procedure is as follows:

1. The DA checks whether there is already address pool configured in the local site when it starts up.
2. The DA will initiate Address Pool request to the CASM Coordinator. It can carry its desired size of address pool in the request, or just use a default value. The address pool size in the DA's request is only used as a hint. The actual size of the address pool is totally determined by CASM Coordinator. It will also carry the DA's identification, the type of transition mechanism and the indication of port allocation support.
3. The CASM Coordinator determines the address pool allocated for the DA based on the parameters received.
4. The CASM Coordinator sends the Address Pool Reply to the DA. It will also distribute the routing entry of the address pool automatically. In particular, if the newly received address pool can be aggregated to an existing one, the routing should be aggregated accordingly.

5.2.3. Interface modeling requirements/Address Pool Status Report

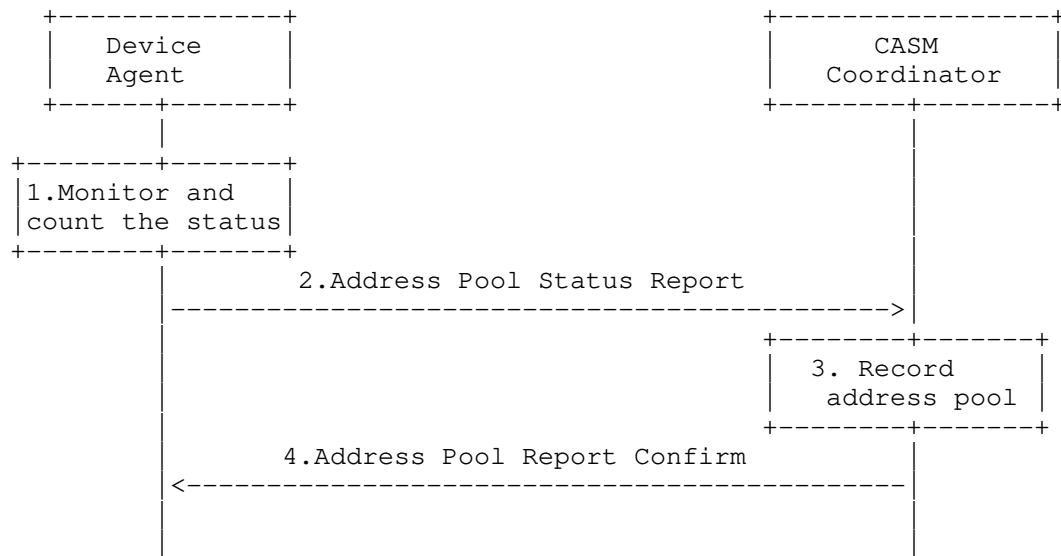


Figure 3: Address Pool Status Report

Figure 3 illustrates the active address pool status report procedure:

1. The DA will monitor and count the usage status of the local address pool. The DA counts the address usage status in one month, one week and one day, which includes the local address, address usage ratio (peak and average values), and the port usage ratio (peak and average values).
2. The DA reports the address pool usage status to the CASM Coordinator. For example, it will report the address usage status in one day, which contains the IP address, NAT44, address list: 30.14.44.0/28, peak address value 14, average address usage ratio 90%, TCP port usage ratio 20%, UDP port usage ratio 30% and etc.
3. The CASM Coordinator records the status and compares with the existing address information to determine whether additional address pool is needed.
4. The CASM Coordinator will confirm the address pool status report request to the DA. It will keep sending the address pool status

report request to the CASM Coordinator if no confirm message is received.

5.2.4. Interface modeling requirements/Address Pool Status Query

When the status of CASM Coordinator is lost or the CASM Coordinator needs the status information of the DAs, the CASM Coordinator may actively query the TD for the status information, as shown in step 1 of Figure 4. The following steps 2,3,4,5 are the same as the Address Pool Status Report procedure.

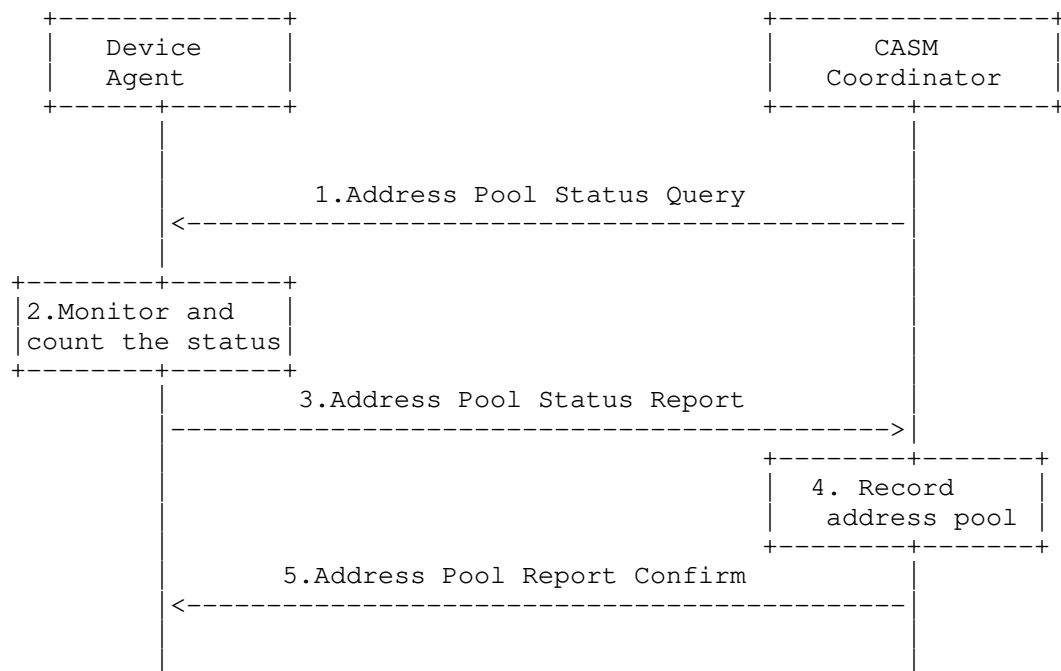


Figure 4: Address Pool Status Query

5.2.5. Interface modeling requirements/Address Exhaustion

When the addresses used by the DA reaches a certain usage threshold, the DA will renew the address pool request to the CASM Coordinator for an additional address pool. The procedure is the same as the initial address pool request.

5.2.6. Interface modeling requirements / Address Pool Release

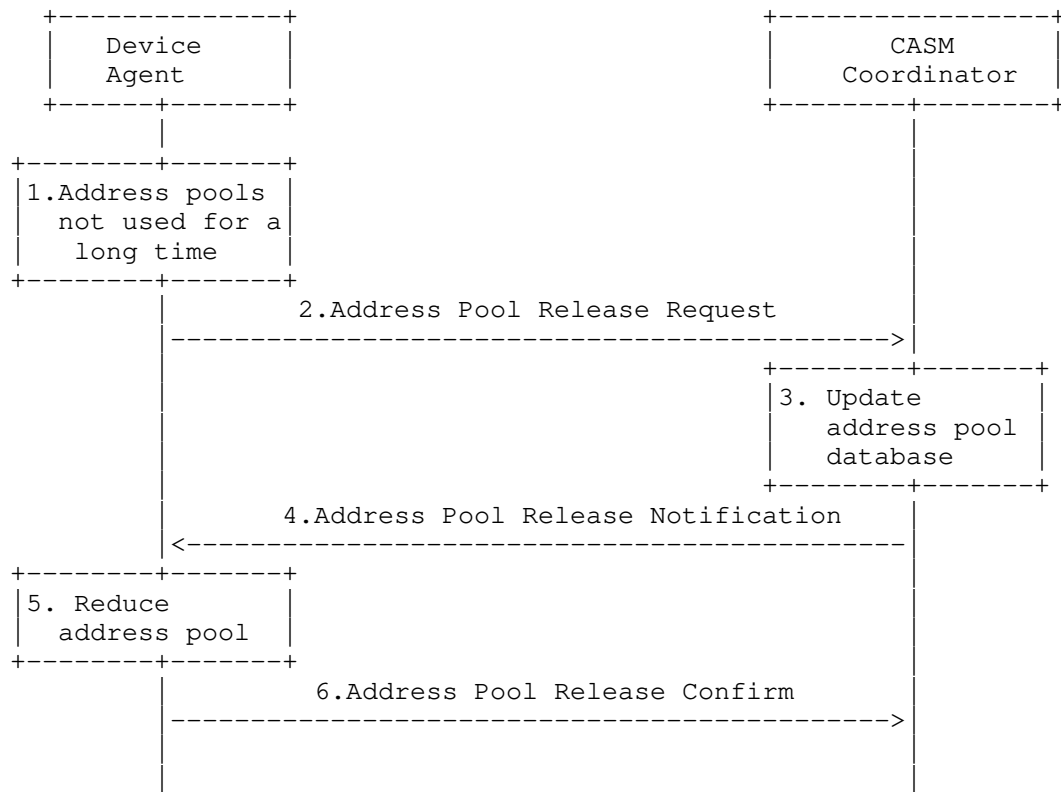


Figure 5: Address Pool Release

Figure 5 illustrates the address pool release procedure:

1. The counting module in the DA checks if the usage threshold of address pool reaches a certain condition;
2. The DA sends the address pool release request to the CASM Coordinator to ask the release of those addresses;
3. The CASM Coordinator updates the local address pool information to add the new addressed released;
4. The CASM Coordinator notifies the TD that the addresses have been release successfully;

5. The DA will update the local address pool. If no Address Pool Release Notification is received, the DA will repeat step 2;
6. Optionally, the DA confirms with the CASM Coordinator that the address pool has been released successfully.

6. Services SDN Management Use Cases

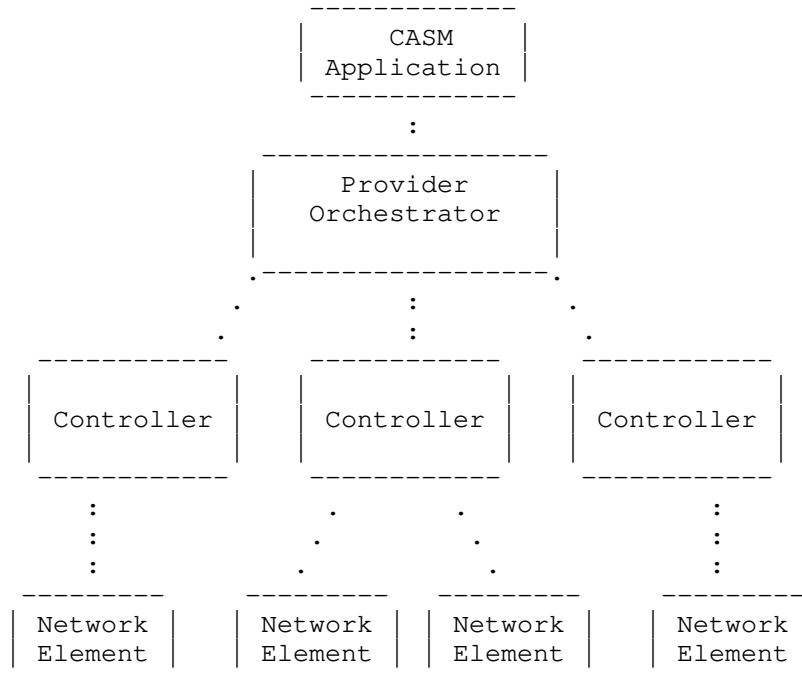


Figure 6: L3 and L2 Services Orchestration

Network Operators need to manage addressing of undelay network elements in order to build end-to-end services and private or public clouds. So address management of customer equipments, provider edges, but also of virtual machines, virtual functions and overlay networks is a very important task. In general the SDN Orchestrators and other management systems must coordinate addressing schemes to ensure network operation. There is need for one address management system that would meet the requirements of such a network deployment. The SDN Orchestrator manages IPv4, IPv6 addresses and also MAC addresses to assign to network interfaces in order to install end-to-end services, and this task can be achieved by the CASM coordination.

A typical use case is the application to the Service provisioning of L3VPN and L2VPN by the SDN orchestration level. For example the architecture presented in [RFC8309] and, more in general in every SDN architecture, could be integrated with CASM. It is important to mention also the possibility of Multi-Provider services, and in this case the two CASM coordinators of the two involved Providers should synchronize. The following Figure shows how CASM Application can communicate with both the Network Operator Orchestrator and, in case of Multi-Provider Service, with another Network Operator Orchestrator too.

7. Security Considerations

8. Acknowledgements

N/A.

9. References

9.1. Normative References

- [RFC2132] Alexander, S. and R. Droms, "DHCP Options and BOOTP Vendor Extensions", RFC 2132, DOI 10.17487/RFC2132, March 1997, <<https://www.rfc-editor.org/info/rfc2132>>.
- [RFC3315] Droms, R., Ed., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, DOI 10.17487/RFC3315, July 2003, <<https://www.rfc-editor.org/info/rfc3315>>.
- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <<https://www.rfc-editor.org/info/rfc6020>>.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <<https://www.rfc-editor.org/info/rfc6241>>.
- [RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <<https://www.rfc-editor.org/info/rfc8040>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

9.2. Informative References

- [RFC6888] Perreault, S., Ed., Yamagata, I., Miyakawa, S., Nakagawa, A., and H. Ashida, "Common Requirements for Carrier-Grade NATs (CGNs)", BCP 127, RFC 6888, DOI 10.17487/RFC6888, April 2013, <<https://www.rfc-editor.org/info/rfc6888>>.

Authors' Addresses

Chen Li
China Telecom
No.118 Xizhimennei street, Xicheng District
Beijing 100035
P.R. China

Email: lichen@ctbri.com.cn

Chongfeng Xie
China Telecom
No.118 Xizhimennei street, Xicheng District
Beijing 100035
P.R. China

Email: xiechf.bri@chinatelecom.cn

Rakesh Kumar
Juniper Networks
1133 Innovation Way
Sunnyvale CA 94089
US

Email: rkkumar@juniper.net

Anil Lohiya
Juniper Networks
1133 Innovation Way
Sunnyvale CA 94089
US

Email: alohiya@juniper.net

Giuseppe Fioccola
Telecom Italia
Via Reiss Romoli, 274
Torino 10148
Italy

Email: giuseppe.fioccola@telecomitalia.it

Weiping Xu
Huawei Technologies
Bantian, Longgang District
shenzhen 518129
P.R. China

Email: xuweiping@huawei.com

Will(Shucheng) Liu
Huawei Technologies
Bantian, Longgang District
shenzhen 518129
P.R. China

Email: liushucheng@huawei.com

Di Ma
ZDNS
4 South 4th St. Zhongguancun
Beijing 100190
P.R. China

Email: madi@zdns.cn

Jun Bi
Tsinghua University
3-212, FIT Building, Tsinghua University, Haidian District
Beijing 100084
P.R. China

Email: junbi@tsinghua.edu.cn

Network Working Group
Internet-Draft
Intended status: Informational
Expires: January 3, 2019

H. Song, Ed.
T. Zhou
ZB. Li
Huawei
G. Fioccola
Telecom Italia
ZQ. Li
China Mobile
P. Martinez-Julia
NICT
L. Ciavaglia
Nokia
A. Wang
China Telecom
July 2, 2018

Toward a Network Telemetry Framework
draft-song-ntf-02

Abstract

This document suggests the necessity of an architectural framework for network telemetry in order to meet the current and future network operation requirements. The defining characteristics of network telemetry shows a clear distinction from the conventional network OAM concept; hence the network telemetry demands new techniques and protocols. This document clarifies the terminologies and classifies the categories and components of a network telemetry framework. The requirements, challenges, existing solutions, and future directions are discussed for each category. The network telemetry framework and the taxonomy help to set a common ground for the collection of related works and put future technique and standard developments into perspective.

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 3, 2019.

Copyright Notice

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 (<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. Motivation	3
1.1. Use Cases	3
1.2. Challenges	5
1.3. Glossary	5
1.4. Network Telemetry	6
2. The Necessity of a Network Telemetry Framework	8
3. Network Telemetry Framework	9
3.1. Existing Works Mapped in the Framework	11
3.2. Management Plane Telemetry	12
3.2.1. Requirements and Challenges	12
3.2.2. Push Extensions for NETCONF	13
3.2.3. gRPC Network Management Interface	13
3.3. Control Plane Telemetry	14
3.3.1. Requirements and Challenges	14
3.3.2. BGP Monitoring Protocol	14
3.4. Data Plane Telemetry	15
3.4.1. Requirements and Challenges	15
3.4.2. Technique Classification	16
3.4.3. The IPFPM technology	16
3.4.4. Dynamic Network Probe	18
3.4.5. IP Flow Information Export (IPFIX) protocol	18

3.4.6. In-Situ OAM	18
3.5. External Data and Event Telemetry	19
3.5.1. Requirements and Challenges	19
4. Security Considerations	20
5. IANA Considerations	20
6. Contributors	20
7. Acknowledgments	20
8. References	20
8.1. Normative References	20
8.2. Informative References	20
Authors' Addresses	23

1. Motivation

The advance of AI/ML technologies gives networks an unprecedented opportunity to realize network autonomy with closed control loops. An intent-driven autonomous network is the logical next step for network evolution following SDN, aiming to reduce (or even eliminate) human labor, make the most efficient use of network resources, and provide better services more aligned with customer requirements. Although we still have a long way to reach the ultimate goal, the journey has started nevertheless.

The storage and computing technologies are already mature enough to be able to retain and process a huge amount of data and make real-time inference. Tools based on machine learning technologies and big data analytics are powerful in detecting and reacting on network faults, anomalies, and policy violations. In turn, the network policy updates for planning, intrusion prevention, optimization, and self-healing can be applied. Some tools can even predict future events based on historical data.

However, the networks fail to keep pace with such data need. The current network architecture, protocol suite, and system design are not ready yet to provide enough quality data. In the remaining of this section, first we identify a few key network operation use cases that network operators need the most. These use cases are also the essential functions of the future autonomous networks. Next, we show why the current network OAM techniques and protocols are not sufficient to meet the requirements of these use cases. The discussion underlines the need of a new brood of techniques and protocols which we put under an umbrella term - network telemetry.

1.1. Use Cases

All these use cases involves the data extracted from the network data plane and sometimes from the network control plane and management plane.

Intent and Policy Compliance: Network policies are the rules that constraint the services for network access, provide differentiate within a service, or enforce specific treatment on the traffic. For example, a service function chain is a policy that requires the selected flows to pass through a set of network functions in order. An intents is a high-level abstract policy which requires a complex translation and mapping process before being applied on networks. While a policy is enforced, the compliance needs to be verified and monitored continuously.

SLA Compliance: A Service-Level Agreement (SLA) defines the level of service a user expects from a network operator, which include the metrics for the service measurement and remedy/penalty procedures when the service level misses the agreement. Users need to check if they get the service as promised and network operators need to evaluate how they can deliver the services that can meet the SLA.

Root Cause Analysis: Network failure often involves a sequence of chained events and the source of the failure is not straightforward to identify, especially when the failure is sporadic. While machine learning or other data analytics technologies can be used for root cause analysis, it up to the network to provide all the relevant data for analysis.

Load Balancing, Traffic Engineering, and Network Planning: Network operators are motivated to optimize their network utilization for better ROI or lower CAPEX, as well as differentiation across services and/or users of a given service. The first step is to know the real-time network conditions before applying policies to steer the user traffic or adjust the load balancing algorithm. In some cases network micro-bursts need to be detected in a very short time-frame so that fine grained traffic control can be applied to avoid possible network congestion. The long term network capacity planning and topology augmentation also rely on the accumulated data of the network operation.

Event Tracking and Prediction: Network visibility is critical for a healthy network operation. Numerous network events are of interest to network operators. For example, Network operators always want to learn where and why packets are dropped for an application flow. They also want to be warned by some early signs that some component is going to fail so the proper fix or replacement can be made in time.

1.2. Challenges

The conventional OAM techniques, as described in [RFC7276], are not sufficient to support the above use cases for the following reasons:

- o Most use cases need to continuously monitor the network and dynamically refine the data collection in real-time and interactively. The poll-based low-frequency data collection is ill-suited for these applications. Streaming data directly pushed from the data source is preferred.
- o Various data is needed from any place ranging from the packet processing engine to the QoS traffic manager. Traditional data plane devices cannot provide the necessary probes. An open and programmable data plane is therefore needed.
- o Many application scenarios need to correlate data from multiple sources (e.g., from distributed nodes or from different network plane). A piecemeal solution is often lacking the capability to consolidate the data from multiple sources. The composition of a complete solution, as partly proposed by ARCA [I-D.pedro-nmrg-anticipated-adaptation], will be empowered and guided by a comprehensive framework.
- o The passive measurement techniques can either consume too much network resources and render too much redundant data, or lead to inaccurate results. The active measurement techniques are indirect, and they can interfere with the user traffic. We need techniques that can collect direct and on-demand data from user traffic.

1.3. Glossary

Before further discussion, we list some key terminology and acronyms used in this documents. We make an intended distinction between network telemetry and network OAM.

AI: Artificial Intelligence. Use machine-learning based technologies to automate network operation.

BMP: BGP Monitoring Protocol

DNP: Dynamic Network Probe

DPI: Deep Packet Inspection

gNMI: gPRC Network Management Interface

gRPC: gRPC Remote Procedure Call

IDN: Intent-Driven Network

IPFIX: IP Flow Information Export Protocol

IPFPM: IP Flow Performance Measurement

IOAM: In-situ OAM

NETCONF: Network Configuration Protocol

Network Telemetry: A general term for a new brood of network visibility techniques and protocols, with the characteristics defined in this document. Network telemetry enables smooth evolution toward intent-driven autonomous networks.

NMS: Network Management System

OAM: Operations, Administration, and Maintenance. A group of network management functions that provide network fault indication, fault localization, performance information, and data and diagnosis functions. Most conventional network monitoring techniques and protocols belong to network OAM.

SNMP: Simple Network Management Protocol

YANG: A data modeling language for NETCONF

YANG FSM: A YANG model to define device side finite state machine

YANG PUSH: A method to subscribe pushed data from remote YANG datastore

1.4. Network Telemetry

For a long time, network operators have relied upon protocols such as SNMP [RFC1157] to monitor the network. SNMP can only provide limited information about the network. Since SNMP is poll-based, it incurs low data rate and high processing overhead. Such drawbacks make SNMP unsuitable for today's automatic network applications.

Network telemetry has emerged as a mainstream technical term to refer to the newer techniques of data collection and consumption, distinguishing itself from the convention techniques for network OAM. It is expected that network telemetry can provide the necessary network visibility for autonomous networks, address the shortcomings

of conventional OAM techniques, and allow for the emergence of new techniques bearing certain characteristics.

One key difference between the network telemetry and the network OAM is that the network telemetry assumes an intelligent machine in the center of a closed control loop, while the network OAM assumes the human network operators in the middle of an open control loop. The network telemetry can directly trigger the automated network operation; The conventional OAM tools only help human operators to monitor and diagnose the networks and guide manual network operations. The different assumptions lead to very different techniques.

Although the network telemetry techniques are just emerging and subject to continuous evolution, several defining characteristics of network telemetry have been well accepted:

- o Push and Streaming: Instead of polling data from network devices, the telemetry collector subscribes to the streaming data pushed from the data source in network devices.
- o Volume and Velocity: The telemetry data is intended to be consumed by machine rather than by human. Therefore, the data volume is huge and the processing is often in realtime.
- o Normalization and Unification: Telemetry aims to address the overall network automation needs. The piecemeal solutions offered by the conventional OAM approach are no longer suitable. Efforts need to be made to normalize the data representation and unify the protocols.
- o Model-based: The data is model-based which allows applications to configure and consume data with ease.
- o Data Fusion: The data for a single application can come from multiple data sources (e.g., cross domain, cross device, and cross layer) and needs to be correlated to take effect.
- o Dynamic and Interactive: Since the network telemetry means to be used in a closed control loop for network automation, it needs to run continuously and adapt to the dynamic and interactive queries from the network operation controller.

In addition, the ideal network telemetry solution should also support the following features:

- o In-Network Customization: The data can be customized in network at run-time to cater to the specific need of applications. This

needs the support of a programmable data plane which allows probes to be deployed at flexible locations.

- o Direct Data Plane Export: The data originated from data plane can be directly exported to the data consumer for efficiency, especially when the data bandwidth is large and the real-time processing is required.
- o In-band Data Collection: In addition to the passive and active data collection approaches, the new hybrid approach allows to directly collect data for any target flow on its entire forwarding path.
- o Non-intrusive: The telemetry system should not fall into the trap of the "observer effect". That is, it should not change the network behavior or affect the forwarding performance.

2. The Necessity of a Network Telemetry Framework

Big data analytics and machine-learning based AI technologies are applied for network operation automation, relying on abundant data from networks. The single-sourced and static data acquisition cannot meet the data requirements. It is desirable to have a framework that integrates multiple telemetry approaches from different layers, and allows flexible combinations for different applications. The framework will benefit application development for the following reasons.

- o The future autonomous networks will require a holistic view on network visibility. All the use cases and applications need to be supported uniformly and coherently under a single intelligent agent. Therefore, the protocols and mechanisms should be consolidated into a minimum yet comprehensive set. A telemetry framework can help to normalize the technique developments.
- o Network visibility presents multiple viewpoints. For example, the device viewpoint takes the network infrastructure as the monitoring object from which the network topology and device status can be acquired; the traffic viewpoint takes the flows or packets as the monitoring object from which the traffic quality and path can be acquired. An application may need to switch its viewpoint during operation. It may also need to correlate a service and its network experience to acquire the comprehensive information.
- o Applications require network telemetry to be elastic in order to efficiently use the network resource and reduce the performance impact. Routine network monitoring covers the entire network with

low data sampling rate. When issues arise or trends emerge, the telemetry data source can be modified and the data rate can be boosted.

- o Efficient data fusion is critical for applications to reduce the overall quantity of data and improve the accuracy of analysis.

So far, some telemetry related work has been done within IETF. However, this work is fragmented and scattered in different working groups. The lack of coherence makes it difficult to assemble a comprehensive network telemetry system and causes repetitive and redundant work.

A formal network telemetry framework is needed for constructing a working system. The framework should cover the concepts and components from the standardization perspective. This document clarifies the layers on which the telemetry is exerted and decomposes the telemetry system into a set of distinct components that the existing and future work can easily map to.

3. Network Telemetry Framework

Telemetry can be applied on the data plane, the control plane, and the management plane in a network, as well as other sources out of the network, as shown in Figure 1.

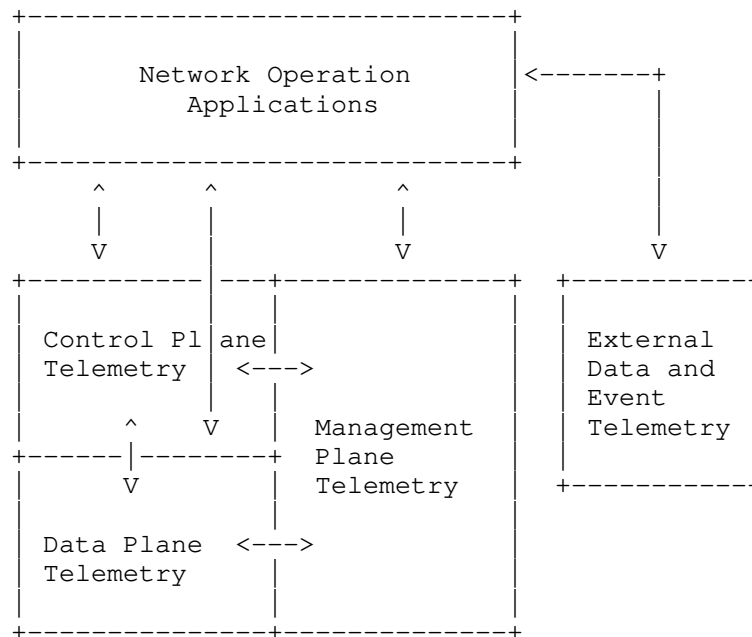


Figure 1: Layer Category of the Network Telemetry Framework

Note that the interaction with the network operation applications can be indirect. For example, in the management plane telemetry, the management plane may need to acquire data from the data plane. On the other hand, an application may involve more than one plane simultaneously. For example, an SLA compliance application may require both the data plane telemetry and the control plane telemetry.

At each plane, the telemetry can be further partitioned into five distinct components:

Data Source: Determine where the original data is acquired. The data source usually just provides raw data which needs further processing. A data source can be considered a probe. A probe can be statically installed or dynamically installed.

Data Subscription: Determine the protocol and channel for applications to acquire desired data. Data subscription is also responsible to define the desired data that might not be directly available from data sources. The subscription data can be described by a model. The model can be statically installed or dynamically installed.

Data Generation: The original data needs to be processed, encoded, and formatted in network devices to meet application subscription requirements. This may involve in-network computing and processing on either the fast path or the slow path in network devices.

Data Export: Determine how the ready data are delivered to applications.

Data Analysis and Storage: In this final step, data is consumed by applications or stored for future reference. Data analysis can be interactive. It may initiate further data subscription.

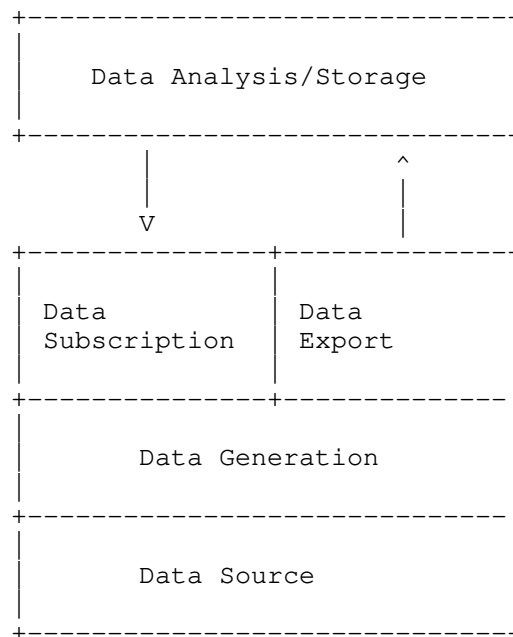


Figure 2: Components in the Network Telemetry Framework

Since most existing standard-related work belongs to the first four components, in the remainder of the document, we focus on these components only.

3.1. Existing Works Mapped in the Framework

The following table provides a non-exhaustive list of existing works (mainly published in IETF and with the emphasis on the latest new technologies) and shows their positions in the framework.

	Management Plane	Control Plane	Data Plane
Data Source	YANG Data Store	Control Proto. Network State	Flow/Packet Statistics States DPI
Data Subscribe	gRPC YANG PUSH	NETCONF/YANG BGP	NETCONF/YANG YANG FSM
Data Generation	Soft DNP	Soft DNP	In-situ OAM IPFPM Hard DNP
Data Export	gRPC YANG PUSH UDP	BMP	IPFIX UDP

Figure 3: Existing Work

3.2. Management Plane Telemetry

3.2.1. Requirements and Challenges

The management plane of the network element interacts with the Network Management System (NMS), and provides information such as performance data, network logging data, network warning and defects data, and network statistics and state data. Some legacy protocols are widely used for the management plane, such as SNMP and Syslog, but these protocols do not meet the requirements of the automatic network operation applications.

New management plane telemetry protocols should consider the following requirements:

Convenient Data Subscription: An application should have the freedom to choose the data export means such as the data types and the export frequency.

Structured Data: For automatic network operation, machines will replace human for network data comprehension. The schema languages such as YANG can efficiently describe structured data and normalize data encoding and transformation.

High Speed Data Transport: In order to retain the information, a server needs to send a large amount of data at high frequency. Compact encoding formats are needed to compress the data and improve the data transport efficiency. The push mode, by replacing the poll mode, can also reduce the interactions between clients and servers, which help to improve the server's efficiency.

3.2.2. Push Extensions for NETCONF

NETCONF [RFC6241] is one popular network management protocol, which is also recommended by IETF. Although it can be used for data collection, NETCONF is good at configurations. YANG Push [I-D.ietf-netconf-yang-push] extends NETCONF and enables subscriber applications to request a continuous, customized stream of updates from a YANG datastore. Providing such visibility into changes made upon YANG configuration and operational objects enables new capabilities based on the remote mirroring of configuration and operational state. Moreover, distributed data collection mechanism [I-D.zhou-netconf-multi-stream-originators] via UDP based publication channel [I-D.ietf-netconf-udp-pub-channel] provides enhanced efficiency for the NETCONF based telemetry.

3.2.3. gRPC Network Management Interface

gRPC Network Management Interface (gNMI) [I-D.openconfig-rtgwg-gnmi-spec] is a network management protocol based on the gRPC [I-D.kumar-rtgwg-grpc-protocol] RPC (Remote Procedure Call) framework. With a single gRPC service definition, both configuration and telemetry can be covered. gRPC is an HTTP/2 [RFC7540] based open source micro service communication framework. It provides a number of capabilities that makes it well-suited for network telemetry, including:

- o Full-duplex streaming transport model combined with a binary encoding mechanism provided further improved telemetry efficiency.
- o gRPC provides higher-level features consistency across platforms that common HTTP/2 libraries typically do not. This characteristic is especially valuable for the fact that telemetry data collectors normally reside on a large variety of platforms.
- o The built-in load-balancing and failover mechanism.

3.3. Control Plane Telemetry

3.3.1. Requirements and Challenges

The control plane telemetry refers to the health condition monitoring of different network protocols, which covers Layer 2 to Layer 7. Keeping track of the running status of these protocols is beneficial for detecting, localizing, and even predicting various network issues, as well as network optimization, in real-time and in fine granularities.

One of the most challenging problems for the control plane telemetry is how to correlate the E2E Key Performance Indicators (KPI) to a specific layer's KPIs. For example, an IPTV user may describe his User Experience (UE) by the video fluency and definition. Then in case of an unusually poor UE KPI or a service disconnection, it is non-trivial work to delimit and localize the issue to the responsible protocol layer (e.g., the Transport Layer or the Network Layer), the responsible protocol (e.g., ISIS or BGP at the Network Layer), and finally the responsible device(s) with specific reasons.

Traditional OAM-based approaches for control plane KPI measurement include PING (L3), Tracert (L3), Y.1731 (L2) and so on. One common issue behind these methods is that they only measure the KPIs instead of reflecting the actual running status of these protocols, making them less effective or efficient for control plane troubleshooting and network optimization. An example of the control plane telemetry is the BGP monitoring protocol (BMP), it is currently used to monitoring the BGP routes and enables rich applications, such as BGP peer analysis, AS analysis, prefix analysis, security analysis, and so on. However, the monitoring of other layers, protocols and the cross-layer, cross-protocol KPI correlations are still in their infancies (e.g., the IGP monitoring is missing), which require substantial further research.

3.3.2. BGP Monitoring Protocol

BGP Monitoring Protocol (BMP) [RFC7854] is used to monitor BGP sessions and intended to provide a convenient interface for obtaining route views.

The BGP routing information is collected from the monitored device(s) to the BMP monitoring station by setting up the BMP TCP session. The BGP peers are monitored by the BMP Peer Up and Peer Down Notifications. The BGP routes (including Adjacency_RIB_In [RFC7854], Adjacency_RIB_out [I-D.ietf-grow-bmp-adj-rib-out], and Local_Rib [I-D.ietf-grow-bmp-local-rib] are encapsulated in the BMP Route Monitoring Message and the BMP Route Mirroring Message, in the form

of both initial table dump and real-time route update. In addition, BGP statistics are reported through the BMP Stats Report Message, which could be either timer triggered or event driven. More BMP extensions can be explored to enrich the applications of BGP monitoring.

3.4. Data Plane Telemetry

3.4.1. Requirements and Challenges

An effective data plane telemetry system relies on the data that the network device can expose. The data's quality, quantity, and timeliness must meet some stringent requirements. This raises some challenges to the network data plane devices where the first hand data originate.

- o A data plane device's main function is user traffic processing and forwarding. While supporting network visibility is important, the telemetry is just an auxiliary function and it should not impede normal traffic processing and forwarding (i.e., the performance is not lowered and the behavior is not altered due to the telemetry functions).
- o The network operation applications requires end-to-end visibility from various sources, which results in a huge volume of data. However, the sheer data quantity should not stress the network bandwidth, regardless of the data delivery approach (i.e., through in-band or out-of-band channels).
- o The data plane devices must provide the data in a timely manner with the minimum possible delay. Long processing, transport, storage, and analysis delay can impact the effectiveness of the control loop and even render the data useless.
- o The data should be structured and labeled, and easy for applications to parse and consume. At the same time, the data types needed by applications can vary significantly. The data plane devices need to provide enough flexibility and programmability to support the precise data provision for applications.
- o The data plane telemetry should support incremental deployment and work even though some devices are unaware of the system. This challenge is highly relevant to the standards and legacy networks.

The industry has agreed that the data plane programmability is essential to support network telemetry. Newer data plane chips are

all equipped with advanced telemetry features and provide flexibility to support customized telemetry functions.

3.4.2. Technique Classification

There can be multiple possible dimensions to classify the data plane telemetry techniques.

Active and Passive: The active and passive methods (as well as the hybrid types) are well documented in [RFC7799]. The passive methods include TCPDUMP, IPFIX [RFC7011], sflow, and traffic mirror. These methods usually have low data coverage. The bandwidth cost is very high in order to improve the data coverage. On the other hand, the active methods include Ping, Traceroute, OWAMP [RFC4656], and TWAMP [RFC5357]. These methods are intrusive and only provide indirect network measurement results. The hybrid methods, including in-situ OAM [I-D.brockners-inband-oam-requirements], IPFPM [RFC8321], and Multipoint Alternate Marking [I-D.fioccola-ippm-multipoint-alt-mark], provide a well-balanced and more flexible approach. However, these methods are also more complex to implement.

In-Band and Out-of-Band: The telemetry data, before being exported to some collector, can be carried in user packets. Such methods are considered in-band (e.g., in-situ OAM [I-D.brockners-inband-oam-requirements]). If the telemetry data is directly exported to some collector without modifying the user packets, Such methods are considered out-of-band (e.g., postcard-based INT). It is possible to have hybrid methods. For example, only the telemetry instruction or partial data is carried by user packets (e.g., IPFPM [RFC8321]).

E2E and In-Network: Some E2E methods start from and end at the network end hosts (e.g., Ping). The other methods work in networks and are transparent to end hosts. However, if needed, the in-network methods can be easily extended into end hosts.

Flow, Path, and Node: Depending on the telemetry objective, the methods can be flow-based (e.g., in-situ OAM [I-D.brockners-inband-oam-requirements]), path-based (e.g., Traceroute), and node-based (e.g., IPFIX [RFC7011]).

3.4.3. The IPFPM technology

The Alternate Marking method is efficient to perform packet loss, delay, and jitter measurements both in an IP and Overlay Networks, as

presented in IPFPM [RFC8321] and [I-D.fioccola-ippm-multipoint-alt-mark].

This technique can be applied to point-to-point and multipoint-to-multipoint flows. Alternate Marking creates batches of packets by alternating the value of 1 bit (or a label) of the packet header. These batches of packets are unambiguously recognized over the network and the comparison of packet counters for each batch allows the packet loss calculation. The same idea can be applied to delay measurement by selecting ad hoc packets with a marking bit dedicated for delay measurements.

Alternate Marking method needs two counters each marking period for each flow under monitor. For instance, by considering n measurement points and m monitored flows, the order of magnitude of the packet counters for each time interval is $n*m*2$ (1 per color).

Since networks offer rich sets of network performance measurement data (e.g packet counters), traditional approaches run into limitations. One reason is the fact that the bottleneck is the generation and export of the data and the amount of data that can be reasonably collected from the network. In addition, management tasks related to determining and configuring which data to generate lead to significant deployment challenges.

Multipoint Alternate Marking approach, described in [I-D.fioccola-ippm-multipoint-alt-mark], aims to resolve this issue and makes the performance monitoring more flexible in case a detailed analysis is not needed.

An application orchestrates network performance measurements tasks across the network to allow an optimized monitoring and it can calibrate how deep can be obtained monitoring data from the network by configuring measurement points roughly or meticulously.

Using Alternate Marking, it is possible to monitor a Multipoint Network without examining in depth by using the Network Clustering (subnetworks that are portions of the entire network that preserve the same property of the entire network, called clusters). So in case there is packet loss or the delay is too high the filtering criteria could be specified more in order to perform a detailed analysis by using a different combination of clusters up to a per-flow measurement as described in IPFPM [RFC8321].

In summary, an application can configure initially an end to end monitoring between ingress points and egress points of the network. If the network does not experiment issues, this approximate monitoring is good enough and is very cheap in terms of network

resources. But, in case of problems, the application becomes aware of the issues from this approximate monitoring and, in order to localize the portion of the network that has issues, configures the measurement points more exhaustively. So a new detailed monitoring is performed. After the detection and resolution of the problem the initial approximate monitoring can be used again.

3.4.4. Dynamic Network Probe

Hardware based Dynamic Network Probe (DNP) [I-D.song-opsawg-dnp4iq] provides a programmable means to customize the data that an application collects from the data plane. A direct benefit of DNP is the reduction of the exported data. A full DNP solution covers several components including data source, data subscription, and data generation. The data subscription needs to define the custom data which can be composed and derived from the raw data sources. The data generation takes advantage of the moderate in-network computing to produce the desired data.

While DNP can introduce unforeseeable flexibility to the data plane telemetry, it also faces some challenges. It requires a flexible data plane that can be dynamically reprogrammed at run-time. The programming API is yet to be defined.

3.4.5. IP Flow Information Export (IPFIX) protocol

Traffic on a network can be seen as a set of flows passing through network elements. IP Flow Information Export (IPFIX) [RFC7011] provides a means of transmitting traffic flow information for administrative or other purposes. A typical IPFIX enabled system includes a pool of Metering Processes collects data packets at one or more Observation Points, optionally filters them and aggregates information about these packets. An Exporter then gathers each of the Observation Points together into an Observation Domain and sends this information via the IPFIX protocol to a Collector.

3.4.6. In-Situ OAM

Traditional passive and active monitoring and measurement techniques are either inaccurate or resource-consuming. It is preferable to directly acquire data associated with a flow's packets when the packets pass through a network. In-situ OAM (iOAM) [I-D.brockners-inband-oam-requirements], a data generation technique, embeds a new instruction header to user packets and the instruction directs the network nodes to add the requested data to the packets. Thus, at the path end the packet's experience on the entire forwarding path can be collected. Such firsthand data is invaluable to many network OAM applications.

However, iOAM also faces some challenges. The issues on performance impact, security, scalability and overhead limits, encapsulation difficulties in some protocols, and cross-domain deployment need to be addressed.

3.5. External Data and Event Telemetry

Events that occur outside the boundaries of the network system are another important source of telemetry information. Correlating both internal telemetry data and external events with the requirements of network systems, as presented in Exploiting External Event Detectors to Anticipate Resource Requirements for the Elastic Adaptation of SDN/NFV Systems [I-D.pedro-nmrg-anticipated-adaptation], provides a strategic and functional advantage to management operations.

3.5.1. Requirements and Challenges

As with other sources of telemetry information, the data and events must meet strict requirements, especially in terms of timeliness, which is essential to properly incorporate external event information to management cycles. Thus, the specific challenges are described as follows:

- o The role of external event detector can be played by multiple elements, including hardware (e.g. physical sensors, such as seismometers) and software (e.g. Big Data sources that analyze streams of information, such as Twitter messages). Thus, the transmitted data must support different shapes but, at the same time, follow a common but extensible ontology.
- o Since the main function of the external event detectors is actually to perform the notifications, their timeliness is assumed. However, once messages have been dispatched, they must be quickly collected and inserted into the control plane with variable priority, which will be high for important sources and/or important events and low for secondary ones.
- o The ontology used by external detectors must be easily adopted by current and future devices and applications. Therefore, it must be easily mapped to current information models, such as in terms of YANG.

Organizing together both internal and external telemetry information will be key for the general exploitation of the management possibilities of current and future network systems, as reflected in the incorporation of cognitive capabilities to new hardware and software (virtual) elements.

4. Security Considerations

TBD

5. IANA Considerations

This document includes no request to IANA.

6. Contributors

The other main contributors of this document are listed as follows.

- o James N. Guichard, Huawei
- o Yunan Gu, Huawei

7. Acknowledgments

We would like to thank Victor Liu and others who have provided helpful comments and suggestions to improve this document.

8. References

8.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.

8.2. Informative References

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

- [I-D.fioccola-ippm-multipoint-alt-mark]
Fioccola, G., Cociglio, M., Sapio, A., and R. Sisto, "Multipoint Alternate Marking method for passive and hybrid performance monitoring", draft-fioccola-ippm-multipoint-alt-mark-04 (work in progress), June 2018.

- [I-D.ietf-grow-bmp-adj-rib-out]
Evens, T., Bayraktar, S., Lucente, P., Mi, K., and S. Zhuang, "Support for Adj-RIB-Out in BGP Monitoring Protocol (BMP)", draft-ietf-grow-bmp-adj-rib-out-01 (work in progress), March 2018.
- [I-D.ietf-grow-bmp-local-rib]
Evens, T., Bayraktar, S., Bhardwaj, M., and P. Lucente, "Support for Local RIB in BGP Monitoring Protocol (BMP)", draft-ietf-grow-bmp-local-rib-01 (work in progress), February 2018.
- [I-D.ietf-netconf-udp-pub-channel]
Zheng, G., Zhou, T., and A. Clemm, "UDP based Publication Channel for Streaming Telemetry", draft-ietf-netconf-udp-pub-channel-03 (work in progress), July 2018.
- [I-D.ietf-netconf-yang-push]
Clemm, A., Voit, E., Prieto, A., Tripathy, A., Nilsen-Nygaard, E., Bierman, A., and B. Lengyel, "YANG Datastore Subscription", draft-ietf-netconf-yang-push-17 (work in progress), July 2018.
- [I-D.kumar-rtgwg-grpc-protocol]
Kumar, A., Kolhe, J., Ghemawat, S., and L. Ryan, "gRPC Protocol", draft-kumar-rtgwg-grpc-protocol-00 (work in progress), July 2016.
- [I-D.openconfig-rtgwg-gnmi-spec]
Shakir, R., Shaikh, A., Borman, P., Hines, M., Lebsack, C., and C. Morrow, "gRPC Network Management Interface (gNMI)", draft-openconfig-rtgwg-gnmi-spec-01 (work in progress), March 2018.
- [I-D.pedro-nmrg-anticipated-adaptation]
Martinez-Julia, P., "Exploiting External Event Detectors to Anticipate Resource Requirements for the Elastic Adaptation of SDN/NFV Systems", draft-pedro-nmrg-anticipated-adaptation-02 (work in progress), June 2018.
- [I-D.song-opsawg-dnp4iq]
Song, H. and J. Gong, "Requirements for Interactive Query with Dynamic Network Probes", draft-song-opsawg-dnp4iq-01 (work in progress), June 2017.

- [I-D.zhou-netconf-multi-stream-originators]
Zhou, T., Zheng, G., Voit, E., Clemm, A., and A. Bierman,
"Subscription to Multiple Stream Originators", draft-zhou-
netconf-multi-stream-originators-02 (work in progress),
May 2018.
- [RFC1157] Case, J., Fedor, M., Schoffstall, M., and J. Davin,
"Simple Network Management Protocol (SNMP)", RFC 1157,
DOI 10.17487/RFC1157, May 1990,
<<https://www.rfc-editor.org/info/rfc1157>>.
- [RFC4656] Shalunov, S., Teitelbaum, B., Karp, A., Boote, J., and M.
Zekauskas, "A One-way Active Measurement Protocol
(OWAMP)", RFC 4656, DOI 10.17487/RFC4656, September 2006,
<<https://www.rfc-editor.org/info/rfc4656>>.
- [RFC5357] Hedayat, K., Krzanowski, R., Morton, A., Yum, K., and J.
Babiarz, "A Two-Way Active Measurement Protocol (TWAMP)",
RFC 5357, DOI 10.17487/RFC5357, October 2008,
<<https://www.rfc-editor.org/info/rfc5357>>.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed.,
and A. Bierman, Ed., "Network Configuration Protocol
(NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011,
<<https://www.rfc-editor.org/info/rfc6241>>.
- [RFC7011] Claise, B., Ed., Trammell, B., Ed., and P. Aitken,
"Specification of the IP Flow Information Export (IPFIX)
Protocol for the Exchange of Flow Information", STD 77,
RFC 7011, DOI 10.17487/RFC7011, September 2013,
<<https://www.rfc-editor.org/info/rfc7011>>.
- [RFC7276] Mizrahi, T., Sprecher, N., Bellagamba, E., and Y.
Weingarten, "An Overview of Operations, Administration,
and Maintenance (OAM) Tools", RFC 7276,
DOI 10.17487/RFC7276, June 2014,
<<https://www.rfc-editor.org/info/rfc7276>>.
- [RFC7540] Belshe, M., Peon, R., and M. Thomson, Ed., "Hypertext
Transfer Protocol Version 2 (HTTP/2)", RFC 7540,
DOI 10.17487/RFC7540, May 2015,
<<https://www.rfc-editor.org/info/rfc7540>>.
- [RFC7799] Morton, A., "Active and Passive Metrics and Methods (with
Hybrid Types In-Between)", RFC 7799, DOI 10.17487/RFC7799,
May 2016, <<https://www.rfc-editor.org/info/rfc7799>>.

- [RFC7854] Scudder, J., Ed., Fernando, R., and S. Stuart, "BGP Monitoring Protocol (BMP)", RFC 7854, DOI 10.17487/RFC7854, June 2016, <<https://www.rfc-editor.org/info/rfc7854>>.
- [RFC8321] Fioccola, G., Ed., Capello, A., Cociglio, M., Castaldelli, L., Chen, M., Zheng, L., Mirsky, G., and T. Mizrahi, "Alternate-Marking Method for Passive and Hybrid Performance Monitoring", RFC 8321, DOI 10.17487/RFC8321, January 2018, <<https://www.rfc-editor.org/info/rfc8321>>.

Authors' Addresses

Haoyu Song (editor)
Huawei
2330 Central Expressway
Santa Clara
USA

Email: haoyu.song@huawei.com

Tianran Zhou
Huawei
156 Beiqing Road
Beijing, 100095
P.R. China

Email: zhoutianran@huawei.com

Zhenbin Li
Huawei
156 Beiqing Road
Beijing, 100095
P.R. China

Email: lizhenbin@huawei.com

Giuseppe Fioccola
Telecom Italia
Via Reiss Romoli, 274
Torino 10148
Italy

Email: giuseppe.fioccola@telecomitalia.it

Zhenqiang Li
China Mobile
No. 32 Xuanwumenxi Ave., Xicheng District
Beijing, 100032
P.R. China

Email: lizhenqiang@chinamobile.com

Pedro Martinez-Julia
NICT
4-2-1, Nukui-Kitamachi
Koganei, Tokyo 184-8795
Japan

Phone: +81 42 327 7293
Email: pedro@nict.go.jp

Laurent Ciavaglia
Nokia
Villardeaux 91460
France

Email: laurent.ciavaglia@nokia.com

Aijun Wang
China Telecom
Beiqijia Town, Changping District
Beijing, 102209
P.R. China

Email: wangaj.bri@chinatelecom.cn

Operations and Management Area Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 4, 2020

Q. Sun
H. Xu
China Telecom
B. Wu, Ed.
Q. Wu, Ed.
Huawei
C. Eckel, Ed.
Cisco Systems
July 3, 2019

A YANG Data Model for SD-WAN Service Delivery
draft-sun-opsawg-sdwan-service-model-04

Abstract

This document provides a YANG data model for an SD-WAN service. An SD-WAN service is a connectivity service offered by a service provider network to provide connectivity across different locations of a customer network or between a customer network and an external network, such as the Internet or a private/public cloud network. This connectivity is provided as an overlay constructed using one of more underlay networks. The model can be used by a service orchestrator of a service provider to request, configure, and manage the components of an SD-WAN service.

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 4, 2020.

Copyright Notice

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

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
1.1. Terminology	3
1.2. Definitions	3
2. High Level Overview of SD-WAN Service	4
3. Service Data Model Usage	6
4. Design of the Data Model	7
4.1. SD-WAN connectivity service	8
4.1.1. VPNs	8
4.1.2. Sites	9
4.2. Application based Policy Service	10
5. Modules Tree Structure	12
6. YANG Modules	17
7. Security Considerations	43
8. IANA Considerations	43
9. Appendix 1: Terminology Mapping between MEF SD-WAN Service Attributes and IETF SD-WAN model	44
10. Appendix 2: IETF OSE model vs IETF SD-WAN model	44
11. Acknowledgments	45
12. Contributors	45
13. References	45
13.1. Normative References	45
13.2. Informative References	46
Authors' Addresses	47

1. Introduction

An SD-WAN service is a connectivity service offered by a service provider network to provide connectivity across different locations of a customer network or between a customer network and an external network. Compared to a conventional PE-based connectivity service as defined in Layer 3 VPN Service Model [RFC8299] and Layer 2 VPN Service Model [RFC8466], an SD-WAN service is a CE-based connectivity service that uses the Internet or PE-based connectivity services as underlay connectivity services. More specially, an SD-WAN service is an overlay connectivity service that provides the flexibility of

adding, removing, or moving services without needing to change the underlay networks.

Besides being an overlay service, an SD-WAN Service has the following characteristics:

- o Hybrid WAN access: The CE could connect to a variety of Internet access technologies, including fiber, cable, DSL-based, WiFi, or 4G/Long Term Evolution (LTE), which implies wider reachability and shorter provisioning cycles. It can also use private VPN connectivity services defined in [RFC4364] and [RFC4664], or Operator Ethernet Services, as defined in [MEF51.1], to take advantage of better performance.
- o Application based traffic forwarding: There are diverse applications used in enterprises, such as VoIP calling, video conferencing, streaming media, etc. Application traffic across the WAN will be forwarded based on business priorities, SLA requirements, or other enterprise requirements.
- o Centralized service management: Subscribers of the service need to be provided a single point (such as a web portal) from which to dynamically add or modify services, such as configuring application policies, adding new sites, or adding new underlay connectivity services.

This draft specifies the SD-WAN service YANG model which is modelled from a customer perspective. The model parameters can be used as an input to automated control and configuration applications to manage SD-WAN services.

1.1. 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].

1.2. Definitions

CE Device: Customer Edge Device , as per Provider Provisioned VPN Terminology [RFC4026] .

CE-based VPN: Refers to Provider Provisioned VPN Terminology [RFC4026]

PE Device: Provider Edge Device, as per Provider Provisioned VPN Terminology [RFC4026]

PE-Based VPNs: Refers to Provider Provisioned VPN Terminology [RFC4026]

SD-WAN: An automated, programmatic approach to managing enterprise network connectivity and circuit usage. It extends software-defined networking (SDN) into an application that businesses can use to quickly create a hybrid WAN, which comprises business-grade IP VPN, broadband Internet, and wireless services or multiple WANs of the same or different types. SD-WAN is also deemed as extended CE-based VPN.

SD-WAN Controller: Refers to the abstract entity that combines Control Plane (CP) and Management Plane (MP) defined in SDN: Layers and Architecture Terminology [RFC7426], to configure, manage and control the CEs and other corresponding SD-WAN components.

Underlay network: A network that provides connectivity across SD-WAN sites and over which customer network packets are tunnelled. An underlay network does not need to be aware that it is carrying overlay customer network packets. Addresses on an underlay network appear as "outer addresses" in encapsulated overlay packets. In general, an underlay network can use a completely different protocol (and address family) from that of the overlay network.

Overlay network: A virtual network in which the separation of customer networks is hidden from the underlying physical infrastructure. That is, the underlying transport networks do not need to know about customer separation to correctly forward traffic. IPsec tunnels [RFC6071] are an example of an L3 overlay network.

2. High Level Overview of SD-WAN Service

From a customer perspective, an example of SD-WAN service network is shown in figure 1.

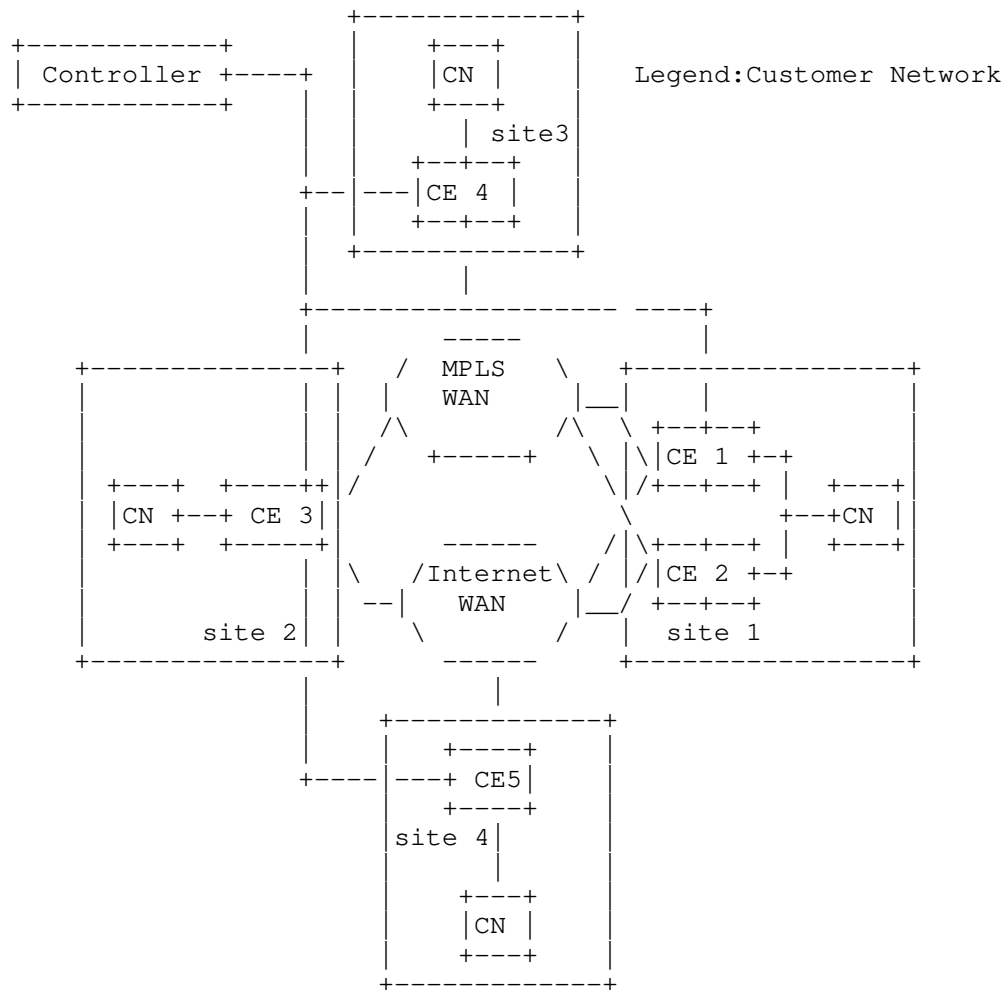


figure 1 SD-WAN network example

As shown in figure 1, the SD-WAN network consists of a number of sites, which are connected through Internet or MPLS VPN.

Within each site, a CE is connected with customer's network on one side, and is also connected to Internet, or to private WAN, or to both on the other side. The customer network could be an L2 or L3 network. For the WAN side, Internet provides ubiquitous IP connectivity via access network like Broadband access or LTE access, while MPLS WAN, like conventional VPN, provides secure and committed connectivity. The boundary between the customer and the service provider is between customer node and the CE device.

Additionally, a site could deploy one or more CEs to improve availability.

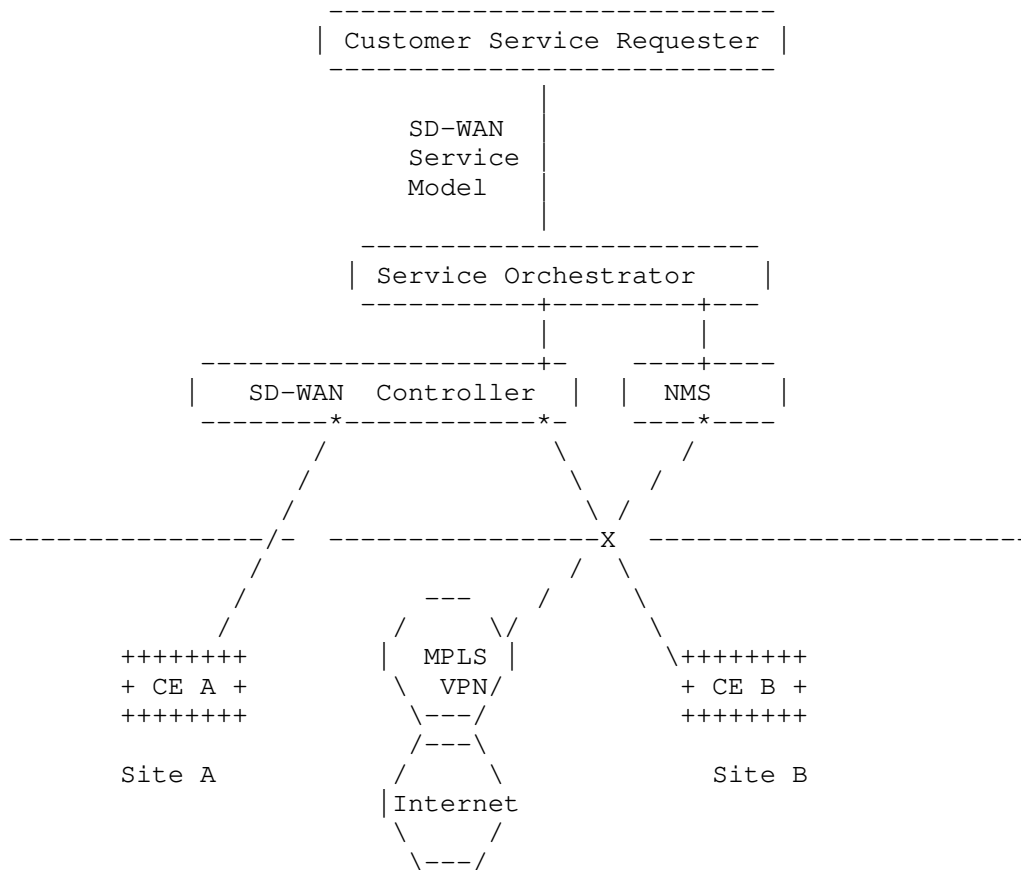
The controller is a centralized entity that manages all the CEs involved in the SD-WAN. The controller could provide bootstrapping of the CEs, ongoing CE configuration, and establishment of secured tunnels between CEs to support the SD-WAN service and application policy enforcement. Various IP tunnelling options (e.g., GRE [RFC2784] and IPSec [RFC6071]), could be used depending on whether traffic from the site is across underlying private VPN or public Internet, and the specific definition is out of scope of this document.

Besides basic connectivity between the sites, the SD-WAN service could be extended by providing direct Internet connectivity, cloud network connectivity, or conventional MPLS VPN interoperability.

3. Service Data Model Usage

The SD-WAN service model provides an abstracted interface to request, configure, and manage the components of an SD-WAN service.

A typical usage for this model is as an input to a service orchestrator that is responsible for service management. Based on the user's service request, the service orchestrator can instruct the SD-WAN controller to add a new site, VPN or application policy in real-time. The orchestrator could orchestrate the other network, such as legacy MPLS VPN network to interconnect with SD-WAN network where Layer 2 VPN Service Mode [RFC8466] or Layer 3 VPN Service Model [RFC8299] could be used.



Reference Architecture for the Use of SD-WAN Service Model Usage

For an SD-WAN to be established under the SP's control, the customer informs the Service Provider of which sites should become part of the requested service and what types of policy will provide. And then the SP configures and updates the service base on the service model and the available resources derived from the SD-WAN controller, and then provisions and manages the customer's service through the SD-WAN controller. How the SD-WAN controller to control and manage the CEs is out of scope of the document.

4. Design of the Data Model

An SD-WAN service consist of two service components:

1. SD-WAN connectivity service

2. SD-WAN application policy service

4.1. SD-WAN connectivity service

SD-WAN connectivity service is the basic component of the SD-WAN service that represents a virtual connection between two or more customer sites. In this model, each virtual connection is defined as a VPN. Each customer can have one or more VPNs, and each VPN can be established between a subset of sites. The association of sites and VPNs is modelled by VPN endpoints.

4.1.1. VPNs

The "sdwan-vpn" list item contains service parameters that apply to an SD-WAN VPN. These parameters are specified as follows:

- o The "vpn-id" leaf is under the vpn-service list, and provides a unique ID for a VPN.
- o The "endpoints" list is under the vpn-service list. Each "endpoint" is a logical point associated with a site. The two main functions of the endpoint are the association of a VPN with a site and per site application based policy enforcement.
- o The "topology" leaf is under the vpn-service list, which refers to a specific topology of the VPN service. Different VPN connection topology can be used. For a VPN with a few sites, simple topologies such as hub-and-spoke or full-mesh can be used. For a large VPN, a hierarchical topology may be taken.
- o The "performance-objectives" container specifies the performance-related properties of an SD-WAN VPN that can be measured. System uptime is the only performance objective defined currently. It indicates the proportion of time, during a given time period that the service is working from the customer perspective. Three parameters are defined, including the start time of the evaluation, the time interval of the evaluation, and the service uptime defined by a percentage.
- o The "reserved-prefixes" container specifies the IP Prefixes that need to be reserved for Service Provider management purposes, such as diagnostics, so as to ensure they are not overlapping with IP Prefixes used by the customer network.

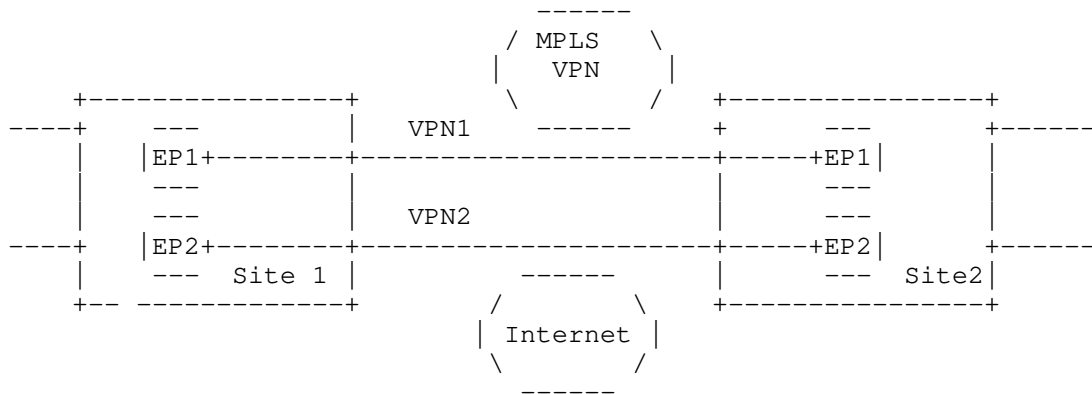


figure 3 SD-WAN VPN example

4.1.2. Sites

A site represents a customer office located at a specific geographic location. The "sites" container specifies the following parameters:

- o "site-id: uniquely identifies the site within the overall network infrastructure.
- o "device" specifies the device type (physical or virtual device) and the number of the devices.
- o "lan-accesses": Specifies the customer network access link parameters. A "site" is composed of at least one "lan-access" where one or more subnets can reside. The "lan-access" consists of the following categories of parameters:
 - * "bearer": defines requirements of the attachment (below Layer 3), bearer type including Ethernet, etc.
 - * IP Connection: defines Layer 3 parameters of the attachment, including IPv4 connection parameters and IPv6 connection parameters.
- o "wan-accesses": Specifies the WAN access link parameters. A "site" is composed of at least one "wan-access". The WAN access can be further specified by access type, service provider name, and bandwidth of the WAN connectivity. The "wan-access" consists of the following categories of parameters:
 - * "access-type": specifies whether the access is Broadband Internet, Wireless Internet or private circuit.

- * "access-provider": specifies the service provider name.
- * bandwidth: specifies the WAN link bandwidth including input and output bandwidth.
- * "bearer": defines requirements of the attachment (below Layer 3), bearer type including Ethernet, etc.
- * IP Connection: defines Layer 3 parameters of the attachment, including IPv4 connection parameters and IPv6 connection parameters.

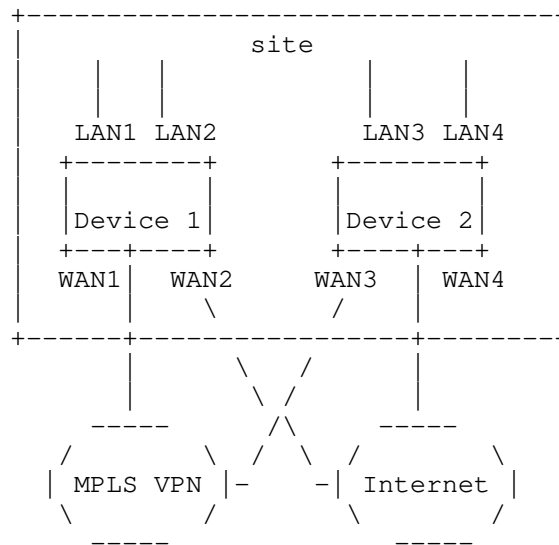


figure 4 Site example

4.2. Application based Policy Service

The connectivity service establishes a virtual connection for the enterprise network, and the Application based Policy Service is designed to ensure business-critical and real-time application experience while also ensuring the security and corporate policies.

Typically, application policies common to each VPN can be defined and then enforced when traffic from a customer's network at a particular site is sent over the WAN.

The application policy assignment is defined under the VPN endpoint container to specify the mapping of application flow name or application group name and their associated policy list names. If an

application flow and the application flow group in which the Application Flow is a member are both assigned a policy at an VPN End Point, the policy assigned to the application flow will supersede the group policy.

The application policy per VPN consists of three lists under the VPN container:

- o application flow list: Describes the characteristics of an enterprise application and is used to identify applications, e.g., based on layer 3 source and destination addresses, layer 4 ports, layer 4 protocol, etc.
- o application group list: Describes application flow aggregation, which is used to deliver aggregation policies, such as bandwidth restrictions for a group of applications.
- o policy list: Defines the application's policy set. Since SD-WAN has more than one WAN connectivity and various encrypted or unencrypted overlay tunnels, there could be multiple tunnel or link selection combination. In this model, different path selection policies are combined to meet different needs based on application SLA, security, cost, and so on. For example, when different applications in a branch need to pass over the WAN, according to the application-aware policy requirements and the IP forwarding table, the Internet application or the SaaS application can be accessed through the Internet, and the data center FTP application can use the Internet encrypted tunnel as the primary path, and the tunnel could only be over broadband Internet instead of wireless internet. This policy combination is not an exhaustive list and could be augmented according to business needs.

An example of a classification of application flows is as follows:

The HTTP traffic from the 192.0.2.0/24 LAN destined for port 80 will be classified in app-id 1.

The FTP traffic from the 192.0.2.0/24 LAN destined for 203.0.113.1/32 will be classified in app-id 2.

An example of a policy list is as follows:

```
"policy": [  
  {  
    "policy-id": "pol-a",  
    "policy-package":  
      {  
        "encryption": "false",  
        "internet-breakout": "true"  
        "public-private": "public",  
        "billing-method": "flat-only"  
        "backup": "false",  
        "bandwidth": "20", "50"  
      }  
    },  
  {  
    "policy-id": "pol-b",  
    "policy-package":  
      {  
        "encryption": "true",  
        "internet-breakout": "false"  
        "public-private": "public",  
        "billing-method": "flat-only"  
        "backup": "false",  
        "bandwidth": "50", "none"  
      }  
    }  
  ]
```

An example of an application policy list is as follows:

```
"app-policy": [  
  {  
    "app-id": "1"  
    "policy-id": "pol-a",  
  },  
  {  
    "app-id": "1"  
    "policy-id": "pol-b",  
  }  
]
```

5. Modules Tree Structure

This document defines an SD-WAN service YANG data model.

```
module: ietf-sdwan-svc  
  +--rw sdwan-svc  
    +--rw vpn-services  
      | +--rw vpn-service* [vpn-id]
```

```

+--rw vpn-id                svc-id
+--rw topology?             identityref
+--rw performance-objective
|   +--rw start-time?        yang:date-and-time
|   +--rw duration?          string
|   +--rw uptime-objective
|       +--rw duration?      decimal64
+--rw reserved-prefixes
|   +--rw prefix*            inet:ip-prefix
+--rw application* [app-id]
|   +--rw app-id             svc-id
|   +--rw ac* [name]
|       +--rw name                                string
|       +--rw (match-type)?
|           +--:(match-flow)
|               +--rw match-flow
|                   +--rw ethertype?                uint16
|                   +--rw cvlan?                     uint8
|                   +--rw ipv4-src-prefix?            inet:ipv4-prefix
|                   +--rw ipv4-dst-prefix?            inet:ipv4-prefix
|                   +--rw l4-src-port?                inet:port-number
|                   +--rw l4-dst-port?                inet:port-number
|                   +--rw ipv6-src-prefix?            inet:ipv6-prefix
|                   +--rw ipv6-dst-prefix?            inet:ipv6-prefix
|                   +--rw protocol-field?             union
|           +--:(match-application)
|               +--rw match-application?             identityref
+--rw application-group* [app-group-id]
|   +--rw app-group-id       svc-id
|   +--rw app-id*            -> ../../application/app-id
+--rw policy* [policy-id]
|   +--rw policy-id          svc-id
|   +--rw policy-package
|       +--rw encryption?    enumeration
|       +--rw public-private? enumeration
|       +--rw local-breakout? boolean
|       +--rw billing-method? enumeration
|       +--rw backup-path?   enumeration
|       +--rw bandwidth
|           +--rw commit?    uint32
|           +--rw max?       uint32
+--rw endpoints* [endpoint-id]
|   +--rw endpoint-id        svc-id
|   +--rw site-role?         identityref
|   +--rw site-attachment
|       +--rw site-id?       -> /sdwan-svc/sites/site/site-id
+--rw endpoint-policy-map
|   +--rw app-group-policy* [app-group-id]

```

```

|         | +---rw app-group-id    leafref
|         | +---rw policy-id?     leafref
+---rw app-policy* [app-id]
|         | +---rw app-id        leafref
|         | +---rw policy-id?    leafref
+---rw sites
  +---rw site* [site-id]
    +---rw site-id      svc-id
    +---rw device* [name]
      | +---rw name      string
      | +---rw type?    identityref
    +---rw lan-access* [name]
      | +---rw name      string
      | +---rw l2-technology
      |   +---rw l2-type?          identityref
      |   +---rw untagged-interface
      |     | +---rw speed?    uint32
      |     | +---rw mode?     neg-mode
      |   +---rw tagged-interface
      |     | +---rw type?          identityref
      |     | +---rw dot1q-vlan-tagged
      |     |   | +---rw tg-type?    identityref
      |     |   | +---rw cvlan-id    uint16
      |     |   +---rw priority-tagged
      |     |     | +---rw tag-type?    identityref
      |   +---rw l2-mtu?          uint32
    +---rw ip-connection
      +---rw ipv4
        | +---rw address-allocation-type?    identityref
        | +---rw dhcp
        |   | +---rw primary-subnet
        |   |   | +---rw ip-prefix?
        |   |   |   | inet:ipv4-prefix
        |   |   +---rw default-router?      inet:ip-address
        |   |   +---rw provider-addresses*
        |   |     | inet:ipv4-address
        |   |   +---rw subscriber-address?  inet:ip-address
        |   |   +---rw reserved-ip-prefix*  inet:ip-prefix
        |   +---rw secondary-subnet* [ip-prefix]
        |     | +---rw ip-prefix
        |     |   | inet:ipv4-prefix
        |     | +---rw provider-addresses*
        |     |   | inet:ipv4-address
        |     | +---rw reserved-ip-prefix*
        |     |   | inet:ipv4-prefix
        +---rw static
          | +---rw primary-subnet
          |   | +---rw ip-prefix?

```

```

|         inet:ipv4-prefix
|         +--rw default-router?          inet:ip-address
|         +--rw provider-addresses*
|         |         inet:ipv4-address
|         +--rw subscriber-address?      inet:ip-address
|         +--rw reserved-ip-prefix*      inet:ip-prefix
+--rw secondary-subnet* [ip-prefix]
|         +--rw ip-prefix
|         |         inet:ipv4-prefix
|         +--rw provider-addresses*
|         |         inet:ipv4-address
|         +--rw reserved-ip-prefix*
|         |         inet:ipv4-prefix
+--rw ipv6
|         +--rw address-allocation-type? identityref
|         +--rw dhcp
|         |         +--rw subnet* [ip-prefix]
|         |         |         +--rw ip-prefix
|         |         |         |         inet:ipv6-prefix
|         |         +--rw provider-addresses*
|         |         |         inet:ipv6-address
|         |         +--rw reserved-ip-prefix*
|         |         |         inet:ipv6-prefix
|         +--rw slaac
|         |         +--rw subnet* [ip-prefix]
|         |         |         +--rw ip-prefix
|         |         |         |         inet:ipv6-prefix
|         |         +--rw provider-addresses*
|         |         |         inet:ipv6-address
|         |         +--rw reserved-ip-prefix*
|         |         |         inet:ipv6-prefix
|         +--rw static
|         |         +--rw subnet* [ip-prefix]
|         |         |         +--rw ip-prefix
|         |         |         |         inet:ipv6-prefix
|         |         +--rw provider-addresses*
|         |         |         inet:ipv6-address
|         |         +--rw reserved-ip-prefix*
|         |         |         inet:ipv6-prefix
|         +--rw subscriber-address?      inet:ipv6-address
+--rw wan-access* [name]
|         +--rw name                      string
|         +--rw access-type?              identityref
|         +--rw access-provider?          string
|         +--rw bandwidth
|         |         +--rw input-bandwidth? uint64
|         |         +--rw output-bandwidth? uint64
+--rw l2-technology

```



```

+--rw l2-type?                identityref
+--rw untagged-interface
|   +--rw speed?      uint32
|   +--rw mode?       neg-mode
+--rw tagged-interface
|   +--rw type?                identityref
|   +--rw dot1q-vlan-tagged
|   |   +--rw tg-type?        identityref
|   |   +--rw cvlan-id        uint16
|   +--rw priority-tagged
|   |   +--rw tag-type?        identityref
+--rw l2-mtu?                  uint32
+--rw ip-connection
+--rw ipv4
|   +--rw address-allocation-type?  identityref
+--rw dhcp
|   +--rw primary-subnet
|   |   +--rw ip-prefix?
|   |   |   inet:ipv4-prefix
|   |   +--rw default-router?        inet:ip-address
|   |   +--rw provider-addresses*
|   |   |   inet:ipv4-address
|   |   +--rw subscriber-address?    inet:ip-address
|   |   +--rw reserved-ip-prefix*    inet:ip-prefix
+--rw secondary-subnet* [ip-prefix]
|   +--rw ip-prefix
|   |   inet:ipv4-prefix
+--rw provider-addresses*
|   inet:ipv4-address
+--rw reserved-ip-prefix*
|   inet:ipv4-prefix
+--rw static
+--rw primary-subnet
|   +--rw ip-prefix?
|   |   inet:ipv4-prefix
+--rw default-router?        inet:ip-address
+--rw provider-addresses*
|   inet:ipv4-address
+--rw subscriber-address?    inet:ip-address
+--rw reserved-ip-prefix*    inet:ip-prefix
+--rw secondary-subnet* [ip-prefix]
|   +--rw ip-prefix
|   |   inet:ipv4-prefix
+--rw provider-addresses*
|   inet:ipv4-address
+--rw reserved-ip-prefix*
|   inet:ipv4-prefix
+--rw ipv6

```

```

+--rw address-allocation-type?  identityref
+--rw dhcp
|   +--rw subnet* [ip-prefix]
|       +--rw ip-prefix
|           |   inet:ipv6-prefix
|       +--rw provider-addresses*
|           |   inet:ipv6-address
|       +--rw reserved-ip-prefix*
|           |   inet:ipv6-prefix
+--rw slaac
|   +--rw subnet* [ip-prefix]
|       +--rw ip-prefix
|           |   inet:ipv6-prefix
|       +--rw provider-addresses*
|           |   inet:ipv6-address
|       +--rw reserved-ip-prefix*
|           |   inet:ipv6-prefix
+--rw static
|   +--rw subnet* [ip-prefix]
|       +--rw ip-prefix
|           |   inet:ipv6-prefix
|       +--rw provider-addresses*
|           |   inet:ipv6-address
|       +--rw reserved-ip-prefix*
|           |   inet:ipv6-prefix
+--rw subscriber-address?  inet:ipv6-address

```

6. YANG Modules

<CODE BEGINS> file "ietf-sdwan-svc@2019-06-06.yang"

```

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

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

  organization
    "IETF foo Working Group.";
  contact
    "WG List: foo@ietf.org
     Editor:  ";

```

```
description
  "The YANG module defines a generic service configuration
  model for Managed SD-WAN.";

revision 2019-06-06 {
  description
    "Initial revision";
  reference "A YANG Data Model for SD-WAN service.";
}

typedef svc-id {
  type string;
  description
    "Type definition for service identifier";
}

typedef address-family {
  type enumeration {
    enum ipv4 {
      description
        "IPv4 address family.";
    }
    enum ipv6 {
      description
        "IPv6 address family.";
    }
  }
  description
    "Defines a type for the address family.";
}

typedef neg-mode {
  type enumeration {
    enum full-duplex {
      description
        "Defining Full duplex mode";
    }
    enum auto-neg {
      description
        "Defining Auto negotiation mode";
    }
  }
  description
    "Defining a type of the negotiation mode";
}

typedef device-type {
  type enumeration {
```

```
    enum physical {
        description
            "Physical device";
    }
    enum virtual {
        description
            "Virtual device";
    }
}
description
    "Defines device types.";
}

identity device-type {
    description
        "Base identity for device type.";
}

identity virtual-ce {
    base device-type;
    description
        "Identity for virtual-ce.";
}

identity physical-ce {
    base device-type;
    description
        "Identity for physical-ce.";
}

identity customer-application {
    description
        "Base identity for customer application.";
}

identity web {
    base customer-application;
    description
        "Identity for Web application (e.g., HTTP, HTTPS).";
}

identity mail {
    base customer-application;
    description
        "Identity for mail application.";
}

identity file-transfer {
```

```
    base customer-application;
    description
        "Identity for file transfer application (e.g., FTP, SFTP).";
}

identity database {
    base customer-application;
    description
        "Identity for database application.";
}

identity social {
    base customer-application;
    description
        "Identity for social-network application.";
}

identity games {
    base customer-application;
    description
        "Identity for gaming application.";
}

identity p2p {
    base customer-application;
    description
        "Identity for peer-to-peer application.";
}

identity network-management {
    base customer-application;
    description
        "Identity for management application
        (e.g., Telnet, syslog, SNMP).";
}

identity voice {
    base customer-application;
    description
        "Identity for voice application.";
}

identity video {
    base customer-application;
    description
        "Identity for video conference application.";
}
```

```
identity eth-inf-type {
  description
    "Identity of the Ethernet interface type.";
}

identity tagged {
  base eth-inf-type;
  description
    "Identity of the tagged interface type.";
}

identity untagged {
  base eth-inf-type;
  description
    "Identity of the untagged interface type.";
}

identity lag {
  base eth-inf-type;
  description
    "Identity of the LAG interface type.";
}

identity tag-type {
  description
    "Base identity from which all tag types
    are derived from";
}

identity c-vlan {
  base tag-type;
  description
    "A Customer-VLAN tag, normally using the 0x8100
    Ethertype";
}

identity tagged-inf-type {
  description
    "Identity for the tagged
    interface type.";
}

identity dot1q {
  base tagged-inf-type;
  description
    "Identity for dot1q vlan tagged interface.";
}
```

```
identity priority-tagged {
  base tagged-inf-type;
  description
    "This identity the priority-tagged interface.";
}

identity vpn-topology {
  description
    "Base identity for vpn topology.";
}

identity any-to-any {
  base vpn-topology;
  description
    "Identity for any-to-any VPN topology.";
}

identity hub-spoke {
  base vpn-topology;
  description
    "Identity for Hub-and-Spoke VPN topology.";
}

identity site-role {
  description
    "Site Role in a VPN topology ";
}

identity any-to-any-role {
  base site-role;
  description
    "Site in an any-to-any IP VPN.";
}

identity hub {
  base site-role;
  description
    "Hub Role in Hub-and-Spoke IP VPN.";
}

identity spoke {
  base site-role;
  description
    "Spoke Role in Hub-and-Spoke IP VPN.";
}

identity access-type {
  description
```

```
    "Access type of a site in a connection to different WAN";
}

identity commodity {
    base access-type;
    description
        "Internet access";
}

identity cellular {
    base access-type;
    description
        "Refers to a subset of 3G/4G/LTE and 5G";
}

identity private {
    base access-type;
    description
        "Refers to private circuits such as Ethernet, T1, etc";
}

identity routing-protocol-type {
    description
        "Base identity for routing protocol type.";
}

identity ospf {
    base routing-protocol-type;
    description
        "Identity for OSPF protocol type.";
}

identity bgp {
    base routing-protocol-type;
    description
        "Identity for BGP protocol type.";
}

identity static {
    base routing-protocol-type;
    description
        "Identity for static routing protocol type.";
}

identity address-allocation-type {
    description
        "Base identity for address-allocation-type for PE-CE link.";
}
```



```
identity dhcp {
  base address-allocation-type;
  description
    "Provider network provides DHCP service to customer.";
}

identity static-address {
  base address-allocation-type;
  description
    "Provider-to-customer addressing is static.";
}

identity slaac {
  base address-allocation-type;
  description
    "Use IPv6 SLAAC.";
}

identity ll-only {
  base address-allocation-type;
  description
    "Use IPv6 Link Local.";
}

identity traffic-direction {
  description
    "Base identity for traffic direction";
}

identity inbound {
  base traffic-direction;
  description
    "Identity for inbound";
}

identity outbound {
  base traffic-direction;
  description
    "Identity for outbound";
}

identity both {
  base traffic-direction;
  description
    "Identity for both";
}

identity traffic-action {
```

```
    description
      "Base identity for traffic action";
  }

  identity permit {
    base traffic-action;
    description
      "Identity for permit action";
  }

  identity deny {
    base traffic-action;
    description
      "Identity for deny action";
  }

  identity bd-limit-type {
    description
      "base identity for bd limit type";
  }

  identity percent {
    base bd-limit-type;
    description
      "Identity for percent";
  }

  identity value {
    base bd-limit-type;
    description
      "Identity for value";
  }

  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.";
```

```
}

grouping vpn-endpoint {
  leaf endpoint-id {
    type svc-id;
    description
      "Identity for the vpn endpoint";
  }
  leaf site-role {
    type identityref {
      base site-role;
    }
    default "any-to-any-role";
    description
      "Role of the site in the VPN.";
  }
  container site-attachment {
    leaf site-id {
      type leafref {
        path "/sdwan-svc/sites/site/site-id";
      }
      description
        "Defines site id attached.";
    }
    description
      "Defines site attachment to a vpn endpoint.";
  }
  container endpoint-policy-map {
    list app-group-policy {
      key "app-group-id";
      leaf app-group-id {
        type leafref {
          path "/sdwan-svc/vpn-services/vpn-service"+
            "/application-group/app-group-id";
        }
        description
          "Identity for application";
      }
      leaf policy-id {
        type leafref {
          path "/sdwan-svc/vpn-services/vpn-service/policy/policy-id";
        }
        description
          "Identity for value";
      }
      description
        "list for application group policy";
    }
  }
}
```

```
list app-policy {
  key "app-id";
  leaf app-id {
    type leafref {
      path "/sdwan-svc/vpn-services/vpn-service"+
        "/application/app-id";
    }
    description
      "Identity for application";
  }
  leaf policy-id {
    type leafref {
      path "/sdwan-svc/vpn-services/vpn-service/policy/policy-id";
    }
    description
      "Identity for value";
  }
  description
    "list for application policy";
}
description
  "Identity for policy maps";
}
description
  "grouping for vpn endpoint";
}

grouping flow-definition {
  container match-flow {
    leaf ethertype {
      type uint16;
      description
        "Ethertype value, e.g. 0800 for IPv4.";
    }
    leaf cvlan {
      type uint8 {
        range "0..7";
      }
      description
        "802.1Q matching.";
    }
    leaf ipv4-src-prefix {
      type inet:ipv4-prefix;
      description
        "Match on IPv4 src address.";
    }
    leaf ipv4-dst-prefix {
      type inet:ipv4-prefix;
    }
  }
}
```

```
        description
            "Match on IPv4 dst address.";
    }
    leaf l4-src-port {
        type inet:port-number;
        description
            "Match on Layer 4 src port.";
    }
    leaf l4-dst-port {
        type inet:port-number;
        description
            "Match on Layer 4 dst port.";
    }
    leaf ipv6-src-prefix {
        type inet:ipv6-prefix;
        description
            "Match on IPv6 src address.";
    }
    leaf ipv6-dst-prefix {
        type inet:ipv6-prefix;
        description
            "Match on IPv6 dst address.";
    }
    leaf protocol-field {
        type union {
            type uint8;
            type identityref {
                base protocol-type;
            }
        }
        description
            "Match on IPv4 protocol or IPv6 Next Header field.";
    }
    description
        "Describes flow-matching criteria.";
}
description
    "Grouping for flow definition.";
}

grouping application-criteria {
    list ac {
        key "name";
        ordered-by user;
        leaf name {
            type string;
            description
                "A description identifying application classification
```

```
        criteria.";
    }
    choice match-type {
        default "match-flow";
        case match-flow {
            uses flow-definition;
        }
        case match-application {
            leaf match-application {
                type identityref {
                    base customer-application;
                }
                description
                    "Defines the application to match.";
            }
        }
        description
            "Choice for classification.";
    }
    description
        "List of marking rules.";
}
description
    "This grouping defines QoS parameters for a site.";
}

grouping vpn-service {
    leaf vpn-id {
        type svc-id;
        description
            "Identity for VPN.";
    }
    leaf topology {
        type identityref {
            base vpn-topology;
        }
        description
            "vpn topology: hub-and-spoke or any-to-any";
    }
    container performance-objective {
        leaf start-time {
            type yang:date-and-time;
            description
                "start-time indicates date and time.";
        }
        leaf duration {
            type string;
            description
```

```
        "Time duration.";
    }
    container uptime-objective {
        leaf duration {
            type decimal64 {
                fraction-digits 5;
                range "0..100";
            }
            units "percent";
            description
                "To be used to define the a percentage of the available
                service.";
        }
        description
            "Uptime objective.";
    }
    description
        "The performance objective.";
}
container reserved-prefixes {
    leaf-list prefix {
        type inet:ip-prefix;
        description
            "ip prefix reserved for SP management purpose.";
    }
    description
        "ip prefix list reserved for SP management purpose.";
}
list application {
    key "app-id";
    leaf app-id {
        type svc-id;
        description
            "application name";
    }
    uses application-criteria;
    description
        "list for application";
}
list application-group {
    key "app-group-id";
    leaf app-group-id {
        type svc-id;
        description
            "application name";
    }
    leaf-list app-id {
        type leafref {
```



```
        path "../../application/app-id";
    }
    description
        "application member list in an application group";
}
description
    "list for application group";
}
list policy {
    key "policy-id";
    leaf policy-id {
        type svc-id;
        description
            "Policy names";
    }
    container policy-package {
        leaf encryption {
            type enumeration {
                enum yes {
                    description
                        "Indicates whether or not the application flow requires
                        to send over encrypted overlay tunnel.";
                }
                enum either {
                    description
                        " Either means this policy is not applied";
                }
            }
        }
        description
            "Indicates whether or not the application flow requires
            encryption.";
    }
    leaf public-private {
        type enumeration {
            enum private-only {
                description
                    "The private WAN underlay is specified.";
            }
            enum either {
                description
                    "Both public WAN or private WAN could be used";
            }
        }
        description
            "Indicates whether the Application Flow can traverse
            Public or Private Underlay Connectivity Services
            (or both).Either means this policy is not applied.";
    }
}
```

```
leaf local-breakout {
    type boolean;
    description
        "indicates whether the Application Flow should be
        routed directly to the Internet using Local Internet
        Breakout.It can have values Yes and No.";
}
leaf billing-method {
    type enumeration {
        enum flat-only {
            description
                "Only flat-rate underlay could be used for the
                traffic.";
        }
        enum either {
            description
                "Either flat-rate or usage based underlay could
                be used for the traffic.";
        }
    }
    description
        "billing policy.";
}
leaf backup-path {
    type enumeration {
        enum yes {
            description
                "Only the primary tunnel overlay could be used for
                the traffic.";
        }
        enum no {
            description
                "Either the primary or backup overlay tunnel could be
                used for the traffic.";
        }
    }
    description
        "overlay connection as Primary or both Primary and
        Backup.";
}
container bandwidth {
    leaf commit {
        type uint32;
        description
            "CIR";
    }
    leaf max {
        type uint32;
    }
}
```

```
        description
            "max speed ";
    }
    description
        "Container for the bandwidth policy";
    }
    description
        "Container for policy package";
    }
    description
        "List for policy";
    }
    list endpoints {
        key "endpoint-id";
        uses vpn-endpoint;
        description
            "List of endpoints.";
    }
    description
        "Grouping of vpn service";
    }

    grouping site-l2-technology {
        container l2-technology {
            leaf l2-type {
                type identityref {
                    base eth-inf-type;
                }
                default "untagged";
                description
                    "Defines physical properties of an interface. By default, the
                     Ethernet interface type is set to 'untagged'.";
            }
            container untagged-interface {
                leaf speed {
                    type uint32;
                    units "mbps";
                    default "10";
                    description
                        "Port speed.";
                }
                leaf mode {
                    type neg-mode;
                    default "auto-neg";
                    description
                        "Negotiation mode.";
                }
            }
            description
```

```
        "Container of Untagged Interface Attributes
        configurations.";
    }
    container tagged-interface {
        leaf type {
            type identityref {
                base tagged-inf-type;
            }
            default "dot1q";
            description
                "Tagged interface type. By default,
                the Tagged interface type is dot1q interface. ";
        }
        container dot1q-vlan-tagged {
            leaf tg-type {
                type identityref {
                    base tag-type;
                }
                default "c-vlan";
                description
                    "TAG type.By default, Tag type is Customer-VLAN tag.";
            }
            leaf cvlan-id {
                type uint16;
                mandatory true;
                description
                    "VLAN identifier.";
            }
            description
                "Tagged interface.";
        }
        container priority-tagged {
            leaf tag-type {
                type identityref {
                    base tag-type;
                }
                default "c-vlan";
                description
                    "TAG type.By default, the TAG type is
                    Customer-VLAN tag.";
            }
            description
                "Priority tagged.";
        }
        description
            "Container for tagged Interface.";
    }
    leaf l2-mtu {
```

```
    type uint32;
    units "bytes";
    description
      " L2 Maximum Frame Size MUST be an integer number of bytes
        >= 1522MTU.";
  }
  description
    "Container for l2 technology.";
}
description
  "grouping for l2 technology.";
}

grouping site-ip-connection {
  container ip-connection {
    container ipv4 {
      leaf address-allocation-type {
        type identityref {
          base address-allocation-type;
        }
        description
          "Defines how addresses are allocated.
            If there is no value for address
            allocation type, then the ipv4 is not enabled.";
      }
    }
    container dhcp {
      container primary-subnet {
        leaf ip-prefix {
          type inet:ipv4-prefix;
          description
            "IPv4 address prefix and mask length between 0 and 31,
              in bits.";
        }
      }
      leaf default-router {
        type inet:ip-address;
        description
          "Address of default router.";
      }
      leaf-list provider-addresses {
        type inet:ipv4-address;
        description
          "the Service Provider IPv4 Addresses MUST be within the
            specified IPv4 Prefix.";
      }
      leaf subscriber-address {
        type inet:ip-address;
        description
          "subscriber IPv4 Addresses: Non-empty list
```

```
        of IPv4 addresses";
    }
    leaf-list reserved-ip-prefix {
        type inet:ip-prefix;
        description
            "List of IPv4 Prefixes, possibly empty";
    }
    description
        "Primary Subnet List";
}
list secondary-subnet {
    key "ip-prefix";
    leaf ip-prefix {
        type inet:ipv4-prefix;
        description
            "IPv4 address prefix and mask length between 0 and 31,
            in bits";
    }
    leaf-list provider-addresses {
        type inet:ipv4-address;
        description
            "Service Provider IPv4 Addresses: Non-empty list
            of IPv4 addresses";
    }
    leaf-list reserved-ip-prefix {
        type inet:ipv4-prefix;
        description
            "List of IPv4 Prefixes, possibly empty";
    }
    description
        "Secondary Subnet List";
}
description
    "DHCP allocated addresses related parameters.";
}
container static {
    container primary-subnet {
        leaf ip-prefix {
            type inet:ipv4-prefix;
            description
                "IPv4 address prefix and mask length between 0 and 31,
                in bits.";
        }
    }
    leaf default-router {
        type inet:ip-address;
        description
            "Address of default router.";
    }
}
```

```
    leaf-list provider-addresses {
      type inet:ipv4-address;
      description
        "the Service Provider IPv4 Addresses MUST be within the
        specified IPv4 Prefix.";
    }
    leaf subscriber-address {
      type inet:ip-address;
      description
        "subscriber IPv4 Addresses: Non-empty list
        of IPv4 addresses";
    }
    leaf-list reserved-ip-prefix {
      type inet:ip-prefix;
      description
        "List of IPv4 Prefixes, possibly empty";
    }
    description
      "Primary Subnet List";
  }
  list secondary-subnet {
    key "ip-prefix";
    leaf ip-prefix {
      type inet:ipv4-prefix;
      description
        "IPv4 address prefix and mask length between 0 and 31,
        in bits";
    }
    leaf-list provider-addresses {
      type inet:ipv4-address;
      description
        "Service Provider IPv4 Addresses: Non-empty list
        of IPv4 addresses";
    }
    leaf-list reserved-ip-prefix {
      type inet:ipv4-prefix;
      description
        "List of IPv4 Prefixes, possibly empty";
    }
    description
      "Secondary Subnet List";
  }
  description
    "Static configuration related parameters.";
}
description
  "IPv4-specific parameters.";
}
```

```
container ipv6 {
  leaf address-allocation-type {
    type identityref {
      base address-allocation-type;
    }
    description
      "Defines how addresses are allocated.
       If there is no value for address
       allocation type, then the ipv6 is not enabled.";
  }
  container dhcp {
    list subnet {
      key "ip-prefix";
      leaf ip-prefix {
        type inet:ipv6-prefix;
        description
          "IPv6 address prefix and prefix length between 0 and
           128";
      }
      leaf-list provider-addresses {
        type inet:ipv6-address;
        description
          "Non-empty list of IPv6 addresses";
      }
      leaf-list reserved-ip-prefix {
        type inet:ipv6-prefix;
        description
          "List of IPv6 Prefixes, possibly empty";
      }
      description
        "Subnet List";
    }
    description
      "DHCP allocated addresses related parameters.";
  }
  container slaac {
    list subnet {
      key "ip-prefix";
      leaf ip-prefix {
        type inet:ipv6-prefix;
        description
          "IPv6 address prefix and prefix length of 64 ";
      }
      leaf-list provider-addresses {
        type inet:ipv6-address;
        description
          "Non-empty list of IPv6 addresses";
      }
    }
  }
}
```



```
        leaf-list reserved-ip-prefix {
            type inet:ipv6-prefix;
            description
                "List of IPv6 Prefixes, possibly empty";
        }
        description
            "Subnet List";
    }
    description
        "DHCP allocated addresses related parameters.";
}
container static {
    list subnet {
        key "ip-prefix";
        leaf ip-prefix {
            type inet:ipv6-prefix;
            description
                "IPv6 address prefix and prefix length between 0 and
                128";
        }
        leaf-list provider-addresses {
            type inet:ipv6-address;
            description
                "Non-empty list of IPv6 addresses";
        }
        leaf-list reserved-ip-prefix {
            type inet:ipv6-prefix;
            description
                "List of IPv6 Prefixes, possibly empty";
        }
        description
            "Subnet List";
    }
    leaf subscriber-address {
        type inet:ipv6-address;
        description
            "IPv6 address or Not Specified.";
    }
    description
        "Static configuration related parameters.";
}
description
    "Describes IPv6 addresses used.";
}
description
    "IPv6-specific parameters.";
}
description
```

```
    "This grouping defines IP connection parameters.";
}

container sdwan-svc {
  container vpn-services {
    list vpn-service {
      key "vpn-id";
      uses vpn-service;
      description
        "List for SD-WAN";
    }
    description
      "Container for SD-WAN VPN service";
  }
  container sites {
    list site {
      key "site-id";
      leaf site-id {
        type svc-id;
        description
          "Site Name";
      }
    }
    list device {
      key "name";
      leaf name {
        type string;
        description
          "Device Name";
      }
      leaf type {
        type identityref {
          base device-type;
        }
        description
          "Device Type: virtual or physical CE";
      }
      description
        "List for device";
    }
  }
  list lan-access {
    key "name";
    leaf name {
      type string;
      description
        "lan access link name";
    }
    uses site-l2-technology;
    uses site-ip-connection;
  }
}
```

```
        description
            "container for lan access";
    }
    list wan-access {
        key "name";
        leaf name {
            type string;
            description
                "wan access link name";
        }
        leaf access-type {
            type identityref {
                base access-type;
            }
            description
                "Access type: Internet, private VPN or cellular";
        }
        leaf access-provider {
            type string;
            description
                "Specifies the name of provider";
        }
        container bandwidth {
            leaf input-bandwidth {
                type uint64;
                description
                    "input bandwidth";
            }
            leaf output-bandwidth {
                type uint64;
                description
                    "output bandwidth";
            }
            description
                "Container for bandwidth";
        }
        uses site-l2-technology;
        uses site-ip-connection;
        description
            "container for wan access";
    }
    description
        "List for site";
}
description
    "Container for sites";
}
description
```

```
    "Top-level container for the SD-WAN services.";
  }
}
```

<CODE ENDS>

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.

8. IANA Considerations

IANA has assigned a new URI from the "IETF XML Registry" [RFC3688].

```
URI: urn:ietf:params:xml:ns:yang:ietf-sdwan-svc
Registrant Contact: The IESG
XML: N/A; the requested URI is an XML namespace.
```

IANA has recorded a YANG module name in the "YANG Module Names" registry [RFC6020] as follows:

```
Name: ietf-sdwan-svc
Namespace: urn:ietf:params:xml:ns:yang:ietf-sdwan-svc
Prefix: sdwan-svc
Reference: RFC xxxx
```

9. Appendix 1: Terminology Mapping between MEF SD-WAN Service Attributes and IETF SD-WAN model

SD-WAN Service Attributes and Services [MEF70-Draft-R1], defines the SD-WAN service attributes and services for SD-WAN service delivery. These service attributes can be used for communication between subscribers and services to deliver SD-WAN services while this draft defines a YANG data model for SD-WAN service delivery communicated between customer and service provider. The purpose of both work is very similar.

The below table shows the terminology mapping. The YANG model retains most parameter definition name but adjusts some of the structure to reserve space for future augmentation. For example, the model defines "vpn-service" and "lan-access" as a list, which can accommodate the case where the current MEF service attribute restricts only one VPN per customer and one LAN access and future extension to multiple VPN or LAN accesses per customer.

IETF SD-WAN Service model	MEF70 R1 SD-WAN Services Term
SD-WAN VPN	SD-WAN Virtual Connection (SWVC)
SD-WAN VPN Endpoint	SWVC End Point
Site	User Network Interface (UNI)
lan-access	UNI link Attributes
wan-access	TBD(Underlay connectivity)

10. Appendix 2: IETF OSE model vs IETF SD-WAN model

SD-WAN OSE service delivery model [I-D.wood-rtgwg-sdwan-ose-yang] 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. Although the OSE YANG model is also a service model instead of being a device model, this model is mainly used for interoperability between multiple SD-WAN domains and service consistency. The differences are shown as follows:

IETF OSE service model	IETF SD-WAN Service model
Domain SD-WAN controller facing	customer-facing
Inter OSE GW connectivity service	unaware of SD-WAN domain in one SP network
Inter SD-WAN domain	Inter-SD-WAN Service Provider TBD
SLA aware dynamic Path selection	static Primary/Backup selection

For the SLA based dynamic path selection policy, the OSE service model uses a similar application classification criteria, but at the same time it will collect the relevant status of the traffic SLA profiles and, based on the measurements calculated from the collected information, the primary or secondary path will be selected.

```

+--primary-backup
  +--rw path-values
    +--rw sla-values
      +--rw latency?          uint32
      +--rw jitter?          uint32
      +--rw packet-loss-rate? uint32

```

11. Acknowledgments

This work has benefited from the discussions of with Jack Pugaczewski, Larry S Samberg, and Pascal Menezes from MEF community.

12. Contributors

The authors would like to thank Zitao Wang for his major contributions to the initial modelling.

13. References

13.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.

13.2. Informative References

- [I-D.wood-rtgwg-sdwan-ose-yang]
Wood, S., Bo, W., Wu, Q., and C. Menezes, "YANG Data Model for SD-WAN OSE service delivery", draft-wood-rtgwg-sdwan-ose-yang-00 (work in progress), March 2019.
- [MEF51.1] MEF, Ed., "Operator Ethernet Service Definition", December 2018, <<https://wiki.mef.net/display/CESG/MEF+51.1+-+OVC+Services>>.
- [MEF70-Draft-R1]
MEF, Ed., "SD-WAN Service Attributes and Services", May 2019, <[https://www.mef.net/Assets/Draft-Standards/MEF_70_Draft_\(R1\).pdf](https://www.mef.net/Assets/Draft-Standards/MEF_70_Draft_(R1).pdf)>.
- [RFC2784] Farinacci, D., Li, T., Hanks, S., Meyer, D., and P. Traina, "Generic Routing Encapsulation (GRE)", RFC 2784, DOI 10.17487/RFC2784, March 2000, <<https://www.rfc-editor.org/info/rfc2784>>.
- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <<https://www.rfc-editor.org/info/rfc3688>>.
- [RFC4026] Andersson, L. and T. Madsen, "Provider Provisioned Virtual Private Network (VPN) Terminology", RFC 4026, DOI 10.17487/RFC4026, March 2005, <<https://www.rfc-editor.org/info/rfc4026>>.
- [RFC4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4364, DOI 10.17487/RFC4364, February 2006, <<https://www.rfc-editor.org/info/rfc4364>>.
- [RFC4664] Andersson, L., Ed. and E. Rosen, Ed., "Framework for Layer 2 Virtual Private Networks (L2VPNs)", RFC 4664, DOI 10.17487/RFC4664, September 2006, <<https://www.rfc-editor.org/info/rfc4664>>.
- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <<https://www.rfc-editor.org/info/rfc6020>>.
- [RFC6071] Frankel, S. and S. Krishnan, "IP Security (IPsec) and Internet Key Exchange (IKE) Document Roadmap", RFC 6071, DOI 10.17487/RFC6071, February 2011, <<https://www.rfc-editor.org/info/rfc6071>>.

- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <<https://www.rfc-editor.org/info/rfc6241>>.
- [RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <<https://www.rfc-editor.org/info/rfc6242>>.
- [RFC7426] Haleplidis, E., Ed., Pentikousis, K., Ed., Denazis, S., Hadi Salim, J., Meyer, D., and O. Koufopavlou, "Software-Defined Networking (SDN): Layers and Architecture Terminology", RFC 7426, DOI 10.17487/RFC7426, January 2015, <<https://www.rfc-editor.org/info/rfc7426>>.
- [RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <<https://www.rfc-editor.org/info/rfc8040>>.
- [RFC8299] Wu, Q., Ed., Litkowski, S., Tomotaki, L., and K. Ogaki, "YANG Data Model for L3VPN Service Delivery", RFC 8299, DOI 10.17487/RFC8299, January 2018, <<https://www.rfc-editor.org/info/rfc8299>>.
- [RFC8341] Bierman, A. and M. Bjorklund, "Network Configuration Access Control Model", STD 91, RFC 8341, DOI 10.17487/RFC8341, March 2018, <<https://www.rfc-editor.org/info/rfc8341>>.
- [RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <<https://www.rfc-editor.org/info/rfc8446>>.
- [RFC8466] Wen, B., Fioccola, G., Ed., Xie, C., and L. Jalil, "A YANG Data Model for Layer 2 Virtual Private Network (L2VPN) Service Delivery", RFC 8466, DOI 10.17487/RFC8466, October 2018, <<https://www.rfc-editor.org/info/rfc8466>>.

Authors' Addresses

Qiong Sun
China Telecom
Beijing
China

Email: sunqiong.bri@chinatelecom.cn

Honglei Xu
China Telecom
Beijing
China

Email: xuhl.bri@chinatelecom.cn

Bo Wu (editor)
Huawei
Nanjing
China

Email: lana.wubo@huawei.com

Qin Wu (editor)
Huawei
Nanjing
China

Email: bill.wu@huawei.com

Charles Eckel (editor)
Cisco Systems
170 W. Tasman Drive
San Jose, CA
United States

Email: eckelcu@cisco.com