Network Working Group
Internet-Draft
Obsoletes: 5798 (if approved)
Intended status: Standards Track
Expires: 10 January 2023

A. Lindem
A. Dogra
Cisco Systems
9 July 2022

Virtual Router Redundancy Protocol (VRRP) Version 3 for IPv4 and IPv6
draft-addogra-rtgwg-vrrp-rfc5798bis-12

Abstract

This document defines the Virtual Router Redundancy Protocol (VRRP) for IPv4 and IPv6. It is version three (3) of the protocol, and it is based on VRRP (version 2) for IPv4 that is defined in RFC 3768 and in "Virtual Router Redundancy Protocol for IPv6". VRRP specifies an election protocol that dynamically assigns responsibility for a virtual router to one of the VRRP routers on a LAN. The VRRP router controlling the IPv4 or IPv6 address(es) associated with a virtual router is called the VRRP Active Router, and it forwards packets sent to these IPv4 or IPv6 addresses. VRRP Active Routers are configured with virtual IPv4 or IPv6 addresses, and VRRP Backup Routers infer the address family of the virtual addresses being carried based on the transport protocol. Within a VRRP router, the virtual routers in each of the IPv4 and IPv6 address families are a domain unto themselves and do not overlap. The election process provides dynamic failover in the forwarding responsibility should the Active Router become unavailable. For IPv4, the advantage gained from using VRRP is a higher-availability default path without requiring configuration of dynamic routing or router discovery protocols on every end-host. For IPv6, the advantage gained from using VRRP for IPv6 is a quicker switchover to Backup Routers than can be obtained with standard IPv6 Neighbor Discovery mechanisms.

The VRRP terminology has been updated conform to inclusive language guidelines for IETF technologies. The IETF has designated National Institute of Standards and Technology (NIST) "Guidance for NIST Staff on Using Inclusive Language in Documentary Standards" for its inclusive language guidelines. This document obsoletes VRRP Version 3 [RFC5798].

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

This document defines the Virtual Router Redundancy Protocol (VRRP) for IPv4 and IPv6. It is version three (3) of the protocol. It is based on VRRP (version 2) for IPv4 that is defined in [RFC3768] and in [VRRP-IPv6]. VRRP specifies an election protocol that dynamically assigns responsibility for a virtual router to one of the VRRP routers on a LAN. The VRRP router controlling the IPv4 or IPv6 address(es) associated with a virtual router is called the VRRP Active Router, and it forwards packets sent to these IPv4 or IPv6 addresses. VRRP Active Routers are configured with virtual IPv4 or IPv6 addresses, and VRRP Backup Routers infer the address family of the virtual addresses being carried based on the transport protocol. Within a VRRP router, the virtual routers in each of the IPv4 and IPv6 address families are a domain unto themselves and do not overlap. The election process provides dynamic failover in the forwarding responsibility should the Active Router become unavailable.

The VRRP terminology has been updated conform to inclusive language guidelines for IETF technologies. The IETF has designated National Institute of Standards and Technology (NIST) "Guidance for NIST Staff on Using Inclusive Language in Documentary Standards" [NISTIR8366] for its inclusive language guidelines. This document obsoletes VRRP Version 3 [RFC5798].

VRRP provides a function similar to the proprietary protocols "Hot Standby Router Protocol (HSRP)" [RFC2281] and "IP Standby Protocol" [IPSTB].

1.1. RFC 5798 Differences

The following changes have been made from RFC 5798:

1. The term for the VRRP router assuming forwarding responsibility has been changed to "Active Router" to be consistent with IETF inclusive terminology. Additionally, inconsistencies in RFC 5798 terminology for both "Active Router" and "Backup Router" were corrected.

2. Errata pertaining to the state machines in Section 12 were corrected.

3. Appendicies describing operation over legacy technologies (FDDI, Token Ring, and ATM LAN Emulation) were removed.

4. Miscellaneous editorial changes were made for readability.

5. The allocation of 224.0.0.18 was added to the IANA considerations section.

2. A Note on Terminology

This document discusses both IPv4 and IPv6 operations, and with respect to the VRRP protocol, many of the descriptions and procedures are common. In this document, it would be less verbose to be able to refer to "IP" to mean either "IPv4 or IPv6". However, historically, the term "IP" usually refers to IPv4. For this reason, in this specification, the term "IPvX" (where X is 4 or 6) is introduced to mean either "IPv4" or "IPv6". In this text, where the IP version matters, the appropriate term is used and the use of the term "IP" is avoided.

3. IPv4

There are a number of methods that an IPv4 end-host can use to determine its first-hop router for a particular IPv4 destination. These include running (or snooping) a dynamic routing protocol such as Routing Information Protocol (RIP) [RFC2453] or OSPF version 2 [RFC2328], running an ICMP router discovery client [RFC1256], or using a statically configured default route.

Running a dynamic routing protocol on every end-host may be infeasible for a number of reasons, including administrative overhead, processing overhead, security issues, or the lack of an implementation for a particular platform. Neighbor or router discovery protocols may require active participation by all hosts on a network, requiring large timer values to reduce protocol overhead associated with the associated protocol packets processing for each host. This can result in a significant delay in the detection of an unreachable router and, such a delay may introduce unacceptably long periods of unreachability for the default route.

The use of a statically configured default route is quite popular since it minimizes configuration and processing overhead on the end-host and is supported by virtually every IPv4 implementation. This mode of operation is likely to persist as dynamic host configuration protocols [RFC2131] are deployed, which typically provide

configuration for an end-host IPv4 address and default gateway. However, this creates a single point of failure. Loss of the default router results in a catastrophic event, isolating all end-hosts that are unable to detect an available alternate path.

The Virtual Router Redundancy Protocol (VRRP) is designed to eliminate the single point of failure inherent in an network utilizing static default routing. VRRP specifies an election protocol that dynamically assigns responsibility for a virtual router to one of the VRRP routers on a LAN. The VRRP router controlling the IPv4 address(es) associated with a virtual router is called the Active Router and forwards packets sent to these IPv4 addresses. The election process provides dynamic failover in the forwarding responsibility should the Active Router become unavailable. Any of the virtual router's IPv4 addresses on a LAN can then be used as the default first hop router by end-hosts. The advantage gained from using VRRP is a higher availability default path without requiring configuration of dynamic routing or router discovery protocols on every end-host.

4. IPv6

IPv6 hosts on a LAN will usually learn about one or more default routers by receiving Router Advertisements sent using the IPv6 Neighbor Discovery (ND) protocol [RFC4861]. The Router Advertisements are multicast periodically at a rate at which the hosts will learn about the default routers in a few minutes. They are not sent frequently enough to rely on the absence of the Router Advertisement to detect router failures.

Neighbor Discovery (ND) includes a mechanism called Neighbor Unreachability Detection to detect the failure of a neighbor node (router or host) or the forwarding path to a neighbor. This is done by sending unicast ND Neighbor Solicitation messages to the neighbor node. To reduce the overhead of sending Neighbor Solicitations, they are only sent to neighbors to which the node is actively sending traffic and only after there has been no positive indication that the router is up for a period of time. Using the default parameters in ND, it will take a host about 38 seconds to learn that a router is unreachable before it will switch to another default router. This delay would be very noticeable to users and cause some transport protocol implementations to time out.

While the ND unreachability detection could be made quicker by changing the parameters to be more aggressive (note that the current lower limit for this is 5 seconds), this would have the downside of significantly increasing the overhead of ND traffic, especially when there are many hosts all trying to determine the reachability of one or more routers.

The Virtual Router Redundancy Protocol for IPv6 provides a much faster switchover to an alternate default router than can be obtained using standard ND procedures. Using VRRP, a Backup Router can take over for a failed default router in around three seconds (using VRRP default parameters). This is done without any interaction with the hosts and a minimum amount of VRRP traffic.

5. Requirements Language

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

6. Scope

The remainder of this document describes the features, design goals, and theory of operation of VRRP. The message formats, protocol processing rules, and state machine that guarantee convergence to a single Active Router are presented. Finally, operational issues related to MAC address mapping, handling of ARP requests, generation of ICMP redirect messages, and security issues are addressed.

7. Definitions

VRRP Router	A router running the Virtual Router Redundancy Protocol. It may participate as one or more virtual routers.
Virtual Router	An abstract object managed by VRRP that acts as a default router for hosts on a shared LAN. It consists of a Virtual Router Identifier and either a set of associated IPv4 addresses or a set of associated IPv6 addresses across a common LAN. A VRRP Router may back up one or more virtual routers.
IP Address Owner	The VRRP router that has the virtual router’s

	IPvX address(es) as real interface address(es). This is the router that, when up, will respond to packets addressed to one of these IPvX addresses for ICMP pings, TCP connection requests, etc.
Primary IP Address	In IPv4, an IPv4 address selected from the set of real interface addresses. One possible selection algorithm is to always select the first address. In IPv4 mode, VRRP advertisements are always sent using the primary IPv4 address as the source of the IPv4 packet. In IPv6, the link-local address of the interface over which the packet is transmitted is used.
Active Router	The VRRP router that is assuming the responsibility of forwarding packets sent to the IPvX address(es) associated with the virtual router, answering ARP requests for the IPv4 address(es), and answering ND requests for the IPv6 address(es). Note that if the IPvX address owner is available, then it will always become the Active Router.
Backup Router(s)	The set of VRRP routers available to assume forwarding responsibility for a virtual router should the current Active Router fail.

8. Required Features

This section describes the set of features that were considered mandatory and that guided the design of VRRP.

8.1. IPvX Address Backup

Backup of an IPvX address or addresses is the primary function of VRRP. When providing election of a Active Router and the additional functionality described below, the protocol should strive to:

* Minimize the duration of unreachability.

* Minimize the steady-state bandwidth overhead and processing complexity.

* Function over a wide variety of multiaccess LAN technologies capable of supporting IPvX traffic.

* Allow multiple virtual routers on a network for load-balancing.

* Support multiple logical IPvX subnets on a single LAN segment.

8.2. Preferred Path Indication

A simple model of Active Router election among a set of redundant routers is to treat each router with equal preference and claim victory after converging to any router as Active Router. However, there are likely to be many environments where there is a distinct preference (or range of preferences) among the set of redundant routers. For example, this preference may be based upon access link cost or speed, router performance or reliability, or other policy considerations. The protocol should allow the expression of this relative path preference in an intuitive manner and guarantee Active Router convergence to the most preferential router currently available.

8.3. Minimization of Unnecessary Service Disruptions

Once Active Router election has been performed, any unnecessary transitions between Active and Backup Routers can result in a disruption in service. The protocol should ensure that, after Active Router election, no state transition is triggered by any Backup Router of equal or lower preference as long as the Active Router continues to function properly.

Some environments may find it beneficial to avoid the state transition triggered when a router that is preferred over the current Active Router becomes available. It may be useful to support an override of the immediate restoration to the preferred path.

8.4. Efficient Operation over Extended LANs

Sending IPvX packets, i.e., sending either IPv4 or IPv6, on a multiaccess LAN requires mapping from an IPvX address to a MAC address. The use of the virtual router MAC address in an extended LAN employing learning bridges can have a significant effect on the bandwidth overhead of packets sent to the virtual router. If the virtual router MAC address is never used as the source address in a link-level frame, then the MAC address location is never learned, resulting in flooding of all packets sent to the virtual router. To improve the efficiency in this environment, the protocol should do the following:

1. Use the virtual router MAC address as the source in a packet sent by the Active Router to trigger MAC learning.

2. Trigger a message immediately after transitioning to the Active Router to update the MAC learning.

3. Trigger periodic messages from the Active Router to maintain the MAC address cache.

8.5. Sub-Second Operation for IPv4 and IPv6

Sub-second detection of Active Router failure is needed in both IPv4 and IPv6 environments. Earlier work proposed that sub-second operation was for IPv6 and this specification leverages that earlier approach for both IPv4 and IPv6.

One possible problematic scenario when using small VRRP_Advertisement_Intervals may occur when a router is generating more packets on a LAN than it can transmit, and a queue builds up on the router. When this occurs, it is possible that packets being transmitted onto the VRRP-protected LAN could see larger queueing delay than the smallest VRRP Advertisement_Interval. In this case, the Active_Down_Interval may be small enough that normal queuing delays might cause a Backup Router to conclude that the Active Router is down, and, hence, promote itself to Active Router. Very shortly afterwards, the delayed VRRP packets from the Active Router cause a switch back to Backup Router. Furthermore, this process can repeat many times per second, causing significant disruption to traffic. To mitigate this problem, priority forwarding of VRRP packets should be considered. The Active Router SHOULD observe that this situation is occurring and log the problem.

9. VRRP Overview

VRRP specifies an election protocol to provide the virtual router function described earlier. All protocol messaging is performed using either IPv4 or IPv6 multicast datagrams. Thus, the protocol can operate over a variety of multiaccess LAN technologies supporting IPvX multicast. Each link of a VRRP virtual router has a single well-known MAC address allocated to it. This document currently only details the mapping to networks using an IEEE 802 48-bit MAC address. The virtual router MAC address is used as the source in all periodic VRRP messages sent by the Active Router to enable MAC learning by layer-2 bridges in an extended LAN.

A virtual router is defined by its virtual router identifier (VRID) and a set of either IPv4 or IPv6 address(es). A VRRP router may associate a virtual router with its real address on an interface. The scope of each virtual router is restricted to a single LAN. A VRRP router may be configured with additional virtual router mappings and priority for virtual routers it is willing to back up. The mapping between the VRID and its IPvX address(es) must be coordinated among all VRRP routers on a LAN.

There is no restriction against reusing a VRID with a different address mapping on different LANs, nor is there a restriction against using the same VRID number for a set of IPv4 addresses and a set of IPv6 addresses. However, these are two different virtual routers.

To minimize network traffic, only the Active Router for each virtual router sends periodic VRRP Advertisement messages. A Backup Router will not attempt to preempt the Active Router unless it has a higher priority. This eliminates service disruption unless a more preferred path becomes available. It’s also possible to administratively prohibit Active Router preemption attempts. The only exception is that a VRRP router will always become the Active Router for any virtual router associated with address(es) it owns. If the Active Router becomes unavailable, then the highest- priority Backup Router will transition to Active Router after a short delay, providing a controlled transition of virtual router responsibility with minimal service interruption.

The VRRP protocol design provides rapid transition from Backup to Active Router to minimize service interruption and incorporates optimizations that reduce protocol complexity while guaranteeing controlled Active Router transition for typical operational scenarios. These optimizations result in an election protocol with minimal runtime state requirements, minimal active protocol states, and a single message type and sender. The typical operational scenarios are defined to be two redundant routers and/or distinct path preferences for each router. A side effect when these assumptions are violated, i.e., more than two redundant paths with equal preference, is that duplicate packets may be forwarded for a brief period during Active Router election. However, the typical scenario assumptions are likely to cover the vast majority of deployments, loss of the Active Router is infrequent, and the expected duration for Active Router election convergence is quite small (< 1 second). Thus, the VRRP optimizations represent significant simplifications in the protocol design while incurring an insignificant probability of brief network disruption.

10. Sample Configurations

10.1. Sample Configuration 1

The following figure shows a simple network with two VRRP routers implementing one virtual router.

```
        +-----------+ +-----------+
        | Router-1  | | Router-2  |
        |(AR VRID=1)| |(BR VRID=1)|
        |           | |           |
VRID=1  +-----------+ +-----------+
IPvX A------>*            *<---------IPvX B
             |            |
             |            |
-------------+------------+--+-----------+-----------+-----------+
                              ^           ^           ^           ^
                              |           |           |           |
     Default Router           |           |           |           |
     IPvX addresses ---> (IPvX A)    (IPvX A)    (IPvX A)    (IPvX A)
                              |           |           |           |
                     IPvX H1->* IPvX  H2->* IPvX  H3->*  IPvX H4->*
                           +--+--+     +--+--+     +--+--+     +--+--+
                           |  H1 |     |  H2 |     |  H3 |     |  H4 |
                           +-----+     +-----+     +--+--+     +--+--+
Legend:
      --+---+---+-- = Ethernet, Token Ring, or FDDI
                  H = Host computer
                 AR = Active Router
                 BR = Backup Router
                 *  =  IPvX Address: X is 4 everywhere in IPv4 case
                                     X is 6 everywhere in IPv6 case
                 (IPvX) = Default Router for hosts
```

In the IPv4 case, i.e., IPvX is IPv4 everywhere in the figure, each router is permanently assigned an IPv4 address on the LAN interface (Router-1 is assigned IPv4 A and Router-2 is assigned IPv4 B), and each host installs a static default route through one of the routers (in this example, they all use Router-1's IPv4 A).

In the IPv6 case, i.e., IPvX is IPv6 everywhere in the figure, each router has its own Link-Local IPv6 address on the LAN interface for the VRRP protocol and a link-local IPv6 address per VRID that is shared with the other routers that serve the same VRID. Each host learns a default route from Router Advertisements through one of the routers (in this example, they all use Router-1's IPv6 Link-Local A).

In an IPv4 VRRP environment, each router supports reception and transmission for the exact same IPv4 address. Router-1 is said to be the IPv4 address owner of IPv4 A, and Router-2 is the IPv4 address

owner of IPv4 B. A virtual router is then defined by associating a unique identifier (the virtual router ID) with the address owned by a router.

In an IPv6 VRRP environment, each router supports reception and transmission with the exact same Link-Local IPv6 address. In an IPv6 VRRP environment, each router will support transmission and reception for the Link-Local IPv6 addresses associated with both VRIDs. Router-1 is said to be the IPv6 address owner of IPv6 A, and Router-2 is the IPv6 address owner of IPv6 B. A virtual router is then defined by associating a unique identifier (the virtual router ID) with the address owned by a router.

Finally, in both the IPv4 and IPv6 cases, the VRRP protocol manages virtual router failover to a Backup Router.

The IPv4 example above shows a virtual router configured to cover the IPv4 address owned by Router-1 (VRID=1, IPv4_Address=A). When VRRP is enabled on Router-1 for VRID=1, it will assert itself as Active Router, with priority = 255, since it is the IP address owner for the virtual router IP address. When VRRP is enabled on Router-2 for VRID=1, it will transition to Backup Router, with priority = 100 (the default priority is 100), since it is not the IPv4 address owner. If Router-1 should fail, then the VRRP protocol will transition Router-2 to Active Router, temporarily taking over forwarding responsibility for IPv4 A to provide uninterrupted service to the hosts. When Router-1 returns to service, it will re-assert itself as Active Router.

The IPv6 example above shows a virtual router configured to cover the IPv6 address owned by Router-1 (VRID=1, IPv6_Address=A). When VRRP is enabled on Router-1 for VRID=1, it will assert itself as Active Router, with priority = 255, since it is the IPv6 address owner for the virtual router IPv6 address. When VRRP is enabled on Router-2 for VRID=1, it will transition to Backup Router, with priority = 100 (the default priority is 100), since it is not the IPv6 address owner. If Router-1 should fail, then the VRRP protocol will transition Router-2 to Active Router, temporarily taking over forwarding responsibility for IPv6 A to provide uninterrupted service to the IPv6 hosts.

Note that in both cases in this example, IPvX B is not backed up and it is only used by Router-2 as its interface address. In order to back up IPvX B, a second virtual router must be configured. This is shown in the next section.

10.2. Sample Configuration 2

The following figure shows a configuration with two virtual routers with the hosts splitting their traffic between them.

```
          +-----------+  +-----------+
          |  Router-1 |  | Router-2  |
          |(AR VRID=1)|  |(BR VRID=1)|
          |(BR VRID=2)|  |(AR VRID=2)|
   VRID=1 +-----------+  +-----------+  VRID=2
   IPvX A ----->*             *<---------- IPvX B
                |             |
                |             |
      ----------+-------------+-+-----------+-----------+-----------+
                                ^           ^           ^           ^
                                |           |           |           |
        Default Router          |           |           |           |
        IPvX addresses ---> (IPvX A)    (IPvX A)    (IPvX A)    (IPvX A)
                                |           |           |           |
                      IPvX H1->* IPvX  H2->* IPvX  H3->*  IPvX H4->*
                            +--+--+     +--+--+     +--+--+     +--+--+
                            |  H1 |     |  H2 |     |  H3 |     |  H4 |
                            +-----+     +-----+     +--+--+     +--+--+

    Legend:
         ---+---+---+--  =  Ethernet, Token Ring, or FDDI
                      H  =  Host computer
                     AR  =  Active Router
                     BR  =  Backup Router
                      *  =  IPvX Address: X is 4 everywhere in IPv4 case
                                           X is 6 everywhere in IPv6 case
                  (IPvX)  =  Default Router for hosts
```

In the IPv4 example above, i.e., IPvX is IPv4 everywhere in the figure, half of the hosts have configured a static default route through Router-1's IPv4 A, and half are using Router-2's IPv4 B. The configuration of virtual router VRID=1 is exactly the same as in the first example (see Section 10.1), and a second virtual router has been added to cover the IPv4 address owned by Router-2 (VRID=2, IPv4_Address=B). In this case, Router-2 will assert itself as Active Router for VRID=2 while Router-1 will act as a Backup Router. This scenario demonstrates a deployment providing load splitting when both routers are available, while providing full redundancy for robustness.

In the IPv6 example above, i.e., IPvX is IPv6 everywhere in the figure, half of the hosts have learned a default route through Router-1's IPv6 A, and half are using Router-2's IPv6 B. The

configuration of virtual router VRID=1 is exactly the same as in the first example (see Section 10.1), and a second virtual router has been added to cover the IPv6 address owned by Router-2 (VRID=2, IPv6_Address=B). In this case, Router-2 will assert itself as Active Router for VRID=2 while Router-1 will act as a Backup Router. This scenario demonstrates a deployment providing load splitting when both routers are available, while providing full redundancy for robustness.

Note that the details of load-balancing are out of scope of this document. However, in a case where the servers need different weights, it may not make sense to rely on router advertisements alone to balance the host traffic between the routers.

11. Protocol

The purpose of the VRRP packet is to communicate to all VRRP routers the priority and the state of the Active Router associated with the VRID.

When VRRP is protecting an IPv4 address, VRRP packets are sent encapsulated in IPv4 packets. They are sent to the IPv4 multicast address assigned to VRRP.

When VRRP is protecting an IPv6 address, VRRP packets are sent encapsulated in IPv6 packets. They are sent to the IPv6 multicast address assigned to VRRP.

11.1. VRRP Packet Format

This section defines the format of the VRRP packet and the relevant fields in the IP header.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    IPv4 Fields or IPv6 Fields                 |
...                                                             ...
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Version| Type  | Virtual Rtr ID|   Priority    |IPvX Addr Count|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   0   |    Max Adver Interval |          Checksum             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                                                               +
|                       IPvX Address(es)                        |
+                                                               +
+                                                               +
+                                                               +
+                                                               +
|                                                               |
+                                                               +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

11.1.1. IPv4 Field Descriptions

11.1.1.1. Source Address

This is the primary IPv4 address of the interface from which the packet is being sent.

11.1.1.2. Destination Address

The IPv4 multicast address as assigned by the IANA for VRRP is:

224.0.0.18

This is a link-local scope multicast address. Routers MUST NOT forward a datagram with this destination address, regardless of its TTL.

11.1.1.3. TTL

The TTL MUST be set to 255. A VRRP router receiving a packet with the TTL not equal to 255 MUST discard the packet.

11.1.1.4. Protocol

The IPv4 protocol number assigned by the IANA for VRRP is 112 (decimal).

11.1.2. IPv6 Field Descriptions

11.1.2.1. Source Address

This is the IPv6 link-local address of the interface from which the packet is being sent.

11.1.2.2. Destination Address

The IPv6 multicast address assigned by the IANA for VRRP is:

```
FF02:0:0:0:0:0:0:12
```

This is a link-local scope multicast address. Routers MUST NOT forward a datagram with this destination address, regardless of its Hop Limit.

11.1.2.3. Hop Limit

The Hop Limit MUST be set to 255. A VRRP router receiving a packet with the Hop Limit not equal to 255 MUST discard the packet.

11.1.2.4. Next Header

The IPv6 Next Header protocol assigned by the IANA for VRRP is 112 (decimal).

11.2. VRRP Field Descriptions

11.2.1. Version

The version field specifies the VRRP protocol version of this packet. This document defines version 3.

11.2.2. Type

The type field specifies the type of this VRRP packet. The only packet type defined in this version of the protocol is:

```
1 - ADVERTISEMENT
```

A packet with unknown type MUST be discarded.

11.2.3. Virtual Rtr ID (VRID)

The Virtual Rtr ID field identifies the virtual router for which this packet is reporting status.

11.2.4. Priority

The priority field specifies the sending VRRP router's priority for the virtual router. Higher values equal higher priority. This field is an 8-bit unsigned integer field.

The priority value for the VRRP router that owns the IPvX address associated with the virtual router MUST be 255 (decimal).

VRRP routers backing up a virtual router MUST use priority values between 1-254 (decimal). The default priority value for VRRP routers backing up a virtual router is 100 (decimal).

The priority value zero (0) has special meaning, indicating that the current Active Router has stopped participating in VRRP. This is used to trigger Backup Routers to quickly transition to Active Router without having to wait for the current Active Router to time out.

11.2.5. IPvX Addr Count

This is the number of either IPv4 addresses or IPv6 addresses contained in this VRRP advertisement. The minimum value is 1.

11.2.6. 0 - Reserved

This reserved field MUST be set to zero on transmission and ignored on reception.

11.2.7. Maximum Advertisement Interval (Max Adver Int)

The Maximum Advertisement Interval is a 12-bit field that indicates the time interval (in centiseconds) between ADVERTISEMENTS. The default is 100 centiseconds (1 second).

Note that higher-priority Active Routers with slower transmission rates than their Backup Routers are unstable. This is because lower-priority nodes configured to faster rates could come online and decide they should be Active Routers before they have heard anything from the higher-priority Active Router with a slower rate. When this happens, it is temporary: once the lower-priority node does hear from the higher-priority Active Router, it will relinquish Active Router status.

11.2.8. Checksum

The checksum field is used to detect data corruption in the VRRP message.

The checksum is the 16-bit one's complement of the one's complement sum of the entire VRRP message starting with the version field and a "pseudo-header" as defined in Section 8.1 of [RFC2460]. The next header field in the "pseudo-header" should be set to 112 (decimal) for VRRP. For computing the checksum, the checksum field is set to zero. See RFC1071 for more detail [RFC1071].

11.2.9. IPvX Address(es)

This refers to one or more IPvX addresses associated with the virtual router. The number of addresses included is specified in the "IP Addr Count" field. These fields are used for troubleshooting misconfigured routers. If more than one address is sent, it is recommended that all routers be configured to send these addresses in the same order to simplify comparisons.

For IPv4 addresses, this refers to one or more IPv4 addresses that are backed up by the virtual router.

For IPv6, the first address must be the IPv6 link-local address associated with the virtual router.

This field contains either one or more IPv4 addresses, or one or more IPv6 addresses. The addresses, IPv4 or IPv6 but not both, MUST be the same as the VRRP protocol packet address family.

12. Protocol State Machine

12.1. Parameters Per Virtual Router

VRID	Virtual Router Identifier. Configurable value in the range 1-255 (decimal). There is no default.
Priority	Priority value to be used by this VRRP router in Active Router election for this virtual router. The value of 255 (decimal) is reserved for the router that owns the IPvX address associated with the virtual router. The value of 0 (zero) is reserved for the Active Router to indicate it is releasing responsibility for the virtual router. The range 1-254

	(decimal) is available for VRRP routers backing up the virtual router. Higher values indicate higher priorities. The default value is 100 (decimal).
IPv4_Addresses	One or more IPv4 addresses associated with this virtual router. Configured list of addresses with no default.
IPv6_Addresses	One or more IPv6 addresses associated with this virtual router. Configured list of addresses with no default. The first address must be the Link-Local address associated with the virtual router.
Advertisement_Interval	Time interval between ADVERTISEMENTS (centiseconds). Default is 100 centiseconds (1 second).
Active_Adver_Interval	Advertisement interval contained in ADVERTISEMENTS received from the Active Router (centiseconds). This value is saved by virtual routers in the Backup state and used to compute Skew_Time and Active_Down_Interval. The initial value is the same as Advertisement_Interval.
Skew_Time	Time to skew Active_Down_Interval in centiseconds. Calculated as: (((256 - priority) * Active_Adver_Interval) / 256)
Active_Down_Interval	Time interval for the Backup Router to declare Active Router down (centiseconds). Calculated as: (3 * Active_Adver_Interval) + Skew_time
Preempt_Mode	Controls whether a (starting or restarting) higher-priority Backup Router preempts a lower-priority Active Router. Values are True to allow preemption and False to prohibit preemption. Default is True.

Note: The exception is that the router that owns the IPvX address associated with the virtual router always preempts, independent of the setting of this flag.

Accept_Mode
: Controls whether a virtual router in Active state will accept packets addressed to the address owner's IPvX address as its own even if it is not the IPvX address owner. The default is False. Deployments that rely on, for example, pinging the address owner's IPvX address may wish to configure Accept_Mode to True.

  Note: IPv6 Neighbor Solicitations and Neighbor Advertisements MUST NOT be dropped when Accept_Mode is False.

Virtual_Router_MAC_Address
: The MAC address used for the source MAC address in VRRP advertisements and advertised in ARP responses as the MAC address to use for IPvX Addresses.

12.2. Timers

Active_Down_Timer
: Timer that fires when a VRRP Advertisement has not been received for Active_Down_Interval.

Adver_Timer
: Timer that fires to trigger transmission of a VRRP Advertisement based on the Advertisement_Interval.

12.3. State Transition Diagram

```
                     +---------------+
          +--------->|               |<-------------+
          |          |  Initialize   |              |
          |   +------|               |----------+   |
          |   |      +---------------+          |   |
          |   |                                 |   |
          |   V                                 V   |
     +---------------+                      +---------------+
     |               |----------------------->|               |
     |    Active     |                      |    Backup     |
     |               |<-----------------------|               |
     +---------------+                      +---------------+
```

12.4. State Descriptions

In the state descriptions below, the state names are identified by {state-name}, and the packets are identified by all-uppercase characters.

A VRRP router implements an instance of the state machine for each virtual router election in which it is participating.

12.4.1. Initialize

The purpose of this state is to wait for a Startup event, that is, an implementation-defined mechanism that initiates the protocol once it has been configured. The configuration mechanism is out of scope of this specification.

```
(100) If a Startup event is received, then:

     (105) - If the Priority = 255, i.e., the router owns the IPvX
             address associated with the virtual router, then:

        (110) + Send an ADVERTISEMENT

        (115) + If the protected IPvX address is an IPv4 address,
                then:

           (120) * For each IPv4 address associated with the virtual
                   router, broadcast a gratuitous ARP request
                   containing the virtual router MAC address and
                   with the target link-layer address set to the
                   virtual router MAC address.

        (125) + else // IPv6

           (130) * For each IPv6 address associated with the virtual
                   router, send an unsolicited ND Neighbor
                   Advertisement with the Router Flag (R) set, the
                   Solicited Flag (S) clear, the Override flag (O)
                   set, the target address set to the IPv6 address
                   of the virtual router, and the target link-layer
                   address set to the virtual router MAC address.

        (135) +endif // was protected address IPv4?

        (140) + Set the Adver_Timer to Advertisement_Interval

        (145) + Transition to the {Active} state

     (150) - else // Router does not own virtual address

        (155) + Set Active_Adver_Interval to Advertisement_Interval

        (160) + Set the Active_Down_Timer to Active_Down_Interval

        (165) + Transition to the {Backup} state

     (170) -endif // priority was not 255

  (175) endif // startup event was received
```

12.5. Backup

The purpose of the {Backup} state is to monitor the availability and state of the Active Router. The Solicited-Node multicast address [RFC4291] is referenced in the pseudo-code below.

```
(300) While in this state, a VRRP router MUST do the following:

      (305) - If the protected IPvX address is an IPv4 address,
              then:

         (310) + MUST NOT respond to ARP requests for the IPv4
                 address(es) associated with the virtual router.

      (315) - else // protected address is IPv6

         (320) + MUST NOT respond to ND Neighbor Solicitation
                 messages for the IPv6 address(es) associated with
                 the virtual router.

         (325) + MUST NOT send ND Router Advertisement messages
                 for the virtual router.

      (330) -endif // was protected address IPv4?

      (335) - MUST discard packets with a destination link-layer
              MAC address equal to the virtual router MAC address.

      (340) - MUST NOT accept packets addressed to the IPvX
              address(es) associated with the virtual router.

      (345) - If a Shutdown event is received, then:

         (350) + Cancel the Active_Down_Timer

         (355) + Transition to the {Initialize} state

      (360) -endif // shutdown received

      (365) - If the Active_Down_Timer fires, then:

         (370) + Send an ADVERTISEMENT

         (375) + If the protected IPvX address is an IPv4 address,
                 then:

            (380) * For each IPv4 address associated with the virtual
                    router, broadcast a gratuitous ARP request
```

```
                        containing the virtual router MAC address and
                        with the target link-layer address set to the
                        virtual router MAC address.

            (385) + else // ipv6

               (390) * Compute and join the Solicited-Node multicast
                       address [RFC4291] for the IPv6 address(es)
                       associated with the virtual router.

               (395) * For each IPv6 address associated with the
                       virtual router, send an unsolicited ND Neighbor
                       Advertisement with the Router Flag (R) set, the
                       Solicited Flag (S) clear, the Override flag (O)
                       set, the target address set to the IPv6 address
                       of the virtual router, and the target link-layer
                       address set to the virtual router MAC address.

            (400) +endif // was protected address ipv4?

            (405) + Set the Adver_Timer to Advertisement_Interval

            (410) + Transition to the {Active} state

         (415) -endif // Active_Down_Timer fired

         (420) - If an ADVERTISEMENT is received, then:

            (425) + If the Priority in the ADVERTISEMENT is zero, then:

               (430) * Set the Active_Down_Timer to Skew_Time

            (440) + else // priority non-zero

               (445) * If Preempt_Mode is False, or if the Priority in
                       the ADVERTISEMENT is greater than or equal to the
                       local Priority, then:

                  (450) @ Set Active_Adver_Interval to Adver Interval
                          contained in the ADVERTISEMENT

                  (455) @ Recompute the Active_Down_Interval

                  (460) @ Reset the Active_Down_Timer to
                          Active_Down_Interval

               (465) * else // preempt was true and priority was less
```

```
                  (470) @ Discard the ADVERTISEMENT

               (475) *endif // preempt test

            (480) +endif // was priority zero?

         (485) -endif // was advertisement received?

      (490) endwhile // Backup state
```

12.6. Active

While in the {Active} state, the router functions as the forwarding router for the IPvX address(es) associated with the virtual router.

Note that in the Active state, the Preempt_Mode Flag is not considered.

```
(600) While in this state, a VRRP router MUST do the following:

      (605) - If the protected IPvX address is an IPv4 address, then:

         (610) + MUST respond to ARP requests for the IPv4
                 address(es) associated with the virtual router.

      (615) - else // IPv6

         (620) + MUST be a member of the Solicited-Node multicast
                 address for the IPv6 address(es) associated with the
                 virtual router.

         (625) + MUST respond to ND Neighbor Solicitation message for
                 the IPv6 address(es) associated with the virtual
                 router.

         (630) + MUST send ND Router Advertisements for the virtual
                 router.

         (635) + If Accept_Mode is False:  MUST NOT drop IPv6
                 Neighbor Solicitations and Neighbor Advertisements.

      (640) +-endif // ipv4?

      (645) - MUST forward packets with a destination link-layer MAC
              address equal to the virtual router MAC address.

      (650) - MUST accept packets addressed to the IPvX address(es)
              associated with the virtual router if it is the IPvX
```

```
                 address owner or if Accept_Mode is True.  Otherwise,
                 MUST NOT accept these packets.

         (655) - If a Shutdown event is received, then:

            (660) + Cancel the Adver_Timer

            (665) + Send an ADVERTISEMENT with Priority = 0

            (670) + Transition to the {Initialize} state

         (675) -endif // shutdown received

         (680) - If the Adver_Timer fires, then:

            (685) + Send an ADVERTISEMENT

            (690) + Reset the Adver_Timer to Advertisement_Interval

         (695) -endif // advertisement timer fired

         (700) - If an ADVERTISEMENT is received, then:

            (705) + If the Priority in the ADVERTISEMENT is zero, then:

               (710) * Send an ADVERTISEMENT

               (715) * Reset the Adver_Timer to Advertisement_Interval

            (720) + else // priority was non-zero

               (725) * If the Priority in the ADVERTISEMENT is greater
                       than the local Priority,

               (730) * or

               (735) * If the Priority in the ADVERTISEMENT is equal to
                       the local Priority and the primary IPvX Address
                       of the sender is greater than the local primary
                       IPvX Address, then:

                  (740) @ Cancel Adver_Timer

                  (745) @ Set Active_Adver_Interval to Adver Interval
                          contained in the ADVERTISEMENT

                  (750) @ Recompute the Skew_Time
```

```
                  (755) @ Recompute the Active_Down_Interval

                  (760) @ Set Active_Down_Timer to Active_Down_Interval

                  (765) @ Transition to the {Backup} state

               (770) * else // new Active Router logic

                  (775) @ Discard ADVERTISEMENT

               (780) *endif // new Active Router detected

            (785) +endif // was priority zero?

         (790) -endif // advert received

      (795) endwhile // in Active state
```

Note: VRRP packets are transmitted with the virtual router MAC address as the source MAC address to ensure that learning bridges correctly determine the LAN segment the virtual router is attached to.

12.7. Virtual Router MAC Address

The virtual router MAC address associated with a virtual router is an IEEE 802 MAC Address in the following format:

IPv4 case: 00-00-5E-00-01-{VRID} (in hex, in Internet-standard bit-order)

The first three octets are derived from the IANA's Organizational Unique Identifier (OUI). The next two octets (00-01) indicate the address block assigned to the VRRP for IPv4 protocol. {VRID} is the VRRP Virtual Router Identifier. This mapping provides for up to 255 IPv4 VRRP routers on a network.

IPv6 case: 00-00-5E-00-02-{VRID} (in hex, in Internet-standard bit-order)

The first three octets are derived from the IANA's OUI. The next two octets (00-02) indicate the address block assigned to the VRRP protocol for the IPv6 protocol. {VRID} is the VRRP Virtual Router Identifier. This mapping provides for up to 255 IPv6 VRRP routers on a network.

12.8. IPv6 Interface Identifiers

IPv6 routers running VRRP MUST create their Interface Identifiers in the normal manner. e.g., "Transmission of IPv6 Packets over Ethernet Networks" [RFC2464]. They MUST NOT use the virtual router MAC address to create the Modified Extended Unique Identifier (EUI)-64 identifiers.

This VRRP specification describes how to advertise and resolve the VRRP router's IPv6 link-local address and other associated IPv6 addresses into the virtual router MAC address.

13. Operational Issues

13.1. IPv4

13.1.1. ICMP Redirects

ICMP redirects may be used normally when VRRP is running between a group of routers. This allows VRRP to be used in environments where the topology is not symmetric.

The IPv4 source address of an ICMP redirect should be the address that the end-host used when making its next-hop routing decision. If a VRRP router is acting as Active Router for virtual router(s) containing addresses it does not own, then it must determine to which virtual router the packet was sent when selecting the redirect source address. One method to deduce the virtual router used is to examine the destination MAC address in the packet that triggered the redirect.

It may be useful to disable redirects for specific cases where VRRP is being used to load-share traffic between a number of routers in a symmetric topology.

13.1.2. Host ARP Requests

When a host sends an ARP request for one of the virtual router IPv4 addresses, the Active Router MUST respond to the ARP request with an ARP response that indicates the virtual MAC address for the virtual router. Note that the source address of the Ethernet frame of this ARP response is the physical MAC address of the physical router. The Active Router MUST NOT respond with its physical MAC address in the ARP response. This allows the client to always use the same MAC address regardless of the current Active Router.

When a VRRP router restarts or boots, it SHOULD NOT send any ARP messages using its physical MAC address for an IPv4 address it owns and, it should only send ARP messages that include virtual MAC addresses.

This may entail the following:

* When configuring an interface, Active Routers should broadcast a gratuitous ARP request containing the virtual router MAC address for each IPv4 address on that interface.

* At system boot, when initializing interfaces for VRRP operation, delay gratuitous ARP requests and ARP responses until both the IPv4 address and the virtual router MAC address are configured.

* When, for example, SSH access to a particular VRRP router is required, an IP address known to belong to that router must be used.

13.1.3. Proxy ARP

If Proxy ARP is to be used on a VRRP router, then the VRRP router must advertise the virtual router MAC address in the Proxy ARP message. Doing otherwise could cause hosts to learn the real MAC address of the VRRP router.

13.2. IPv6

13.2.1. ICMPv6 Redirects

ICMPv6 redirects may be used normally when VRRP is running between a group of routers [RFC4443]. This allows VRRP to be used in environments where the topology is not symmetric, e.g., the VRRP routers do not connect to the same destinations.

The IPv6 source address of an ICMPv6 redirect should be the address that the end-host used when making its next-hop routing decision. If a VRRP router is acting as Active Router for virtual router(s) containing addresses it does not own, then it must determine to which virtual router the packet was sent when selecting the redirect source address. A method to deduce the virtual router used is to examine the destination MAC address in the packet that triggered the redirect.

13.2.2. ND Neighbor Solicitation

When a host sends an ND Neighbor Solicitation message for the virtual router IPv6 address, the Active Router MUST respond to the ND Neighbor Solicitation message with the virtual MAC address for the virtual router. The Active Router MUST NOT respond with its physical MAC address. This allows the client to always use the same MAC address regardless of the current Active Router.

When an Active Router sends an ND Neighbor Solicitation message for a host's IPv6 address, the Active Router MUST include the virtual MAC address for the virtual router if it sends a source link-layer address option in the neighbor solicitation message. It MUST NOT use its physical MAC address in the source link-layer address option.

When a VRRP router restarts or boots, it SHOULD NOT send any ND messages with its physical MAC address for the IPv6 address it owns and, it should only send ND messages that include virtual MAC addresses.

This may entail the following:

* When configuring an interface, Active Routers should send an unsolicited ND Neighbor Advertisement message containing the virtual router MAC address for the IPv6 address on that interface.

* At system boot, when initializing interfaces for VRRP operation, all ND Router and Neighbor Advertisements and Solicitation messages must be delayed until both the IPv6 address and the virtual router MAC address are configured.

Note that on a restarting Active Router where the VRRP protected address is an interface address, i.e., the address owner, duplicate address detection (DAD) may fail, as the Backup Router may answer that it owns the address. One solution is to not run DAD in this case.

13.2.3. Router Advertisements

When a Backup VRRP router has become Active Router for a virtual router, it is responsible for sending Router Advertisements for the virtual router as specified in Section 12.6. The Backup Routers must be configured to send the same Router Advertisement options as the address owner.

Router Advertisement options that advertise special services, e.g., Home Agent Information Option, that are present in the address owner should not be sent by the address owner unless the Backup Routers are prepared to assume these services in full and have a complete and synchronized database for this service.

13.2.4. Unsolicited Neighbor Advertisements

A VRRP router acting as either an IPv6 Active Router or Backup Router, SHOULD accept Unsolicited Neighbor Advertisements and update the corresponding neighbor cache [RFC4861]. Since these are sent to the IPv6 all-nodes multicast address (FF02:1) [RFC4861] or the IPv6 all-routers multicast address (FF02:2), they will be received. Unsolicited Neighbor Advertisements are sent both in the case where the link-level addresses changes [RFC4861] and for gratuitous neighbor discovery by first hop routers [RFC9131]. Additional configuration MAY be required in order for Unsolicited Neighbor Advertisements to update the corresponding neighbor cache.

13.3. IPvX

13.3.1. Potential Forwarding Loop

If it is not the address owner, a VRRP router SHOULD NOT forward packets addressed to the IPvX address for which it becomes Active Router. Forwarding these packets would result in unnecessary traffic. Also, in the case of LANs that receive packets they transmit, e.g., Token Ring, this can result in a forwarding loop that is only terminated when the IPvX TTL expires.

One such mechanism for VRRP routers is to add/delete a reject host route for each adopted IPvX address when transitioning to/from Active state.

13.3.2. Recommendations Regarding Setting Priority Values

A priority value of 255 designates a particular router as the "IPvX address owner". Care must be taken not to configure more than one router on the link in this way for a single VRID.

Routers with priority 255 will, as soon as they start up, preempt all lower-priority routers. No more than one router on the link is to be configured with priority 255, especially if preemption is set. If no router has this priority, and preemption is disabled, then no preemption will occur.

When there are multiple Backup Routers, their priority values should be uniformly distributed. For example, if one Backup Router has the default priority of 100 and another Backup Router is added, a priority of 50 would be a better choice for it than 99 or 100, in order to facilitate faster convergence.

13.4. VRRPv3 and VRRPv2 Interoperation

13.4.1. Assumptions

1. VRRPv2 and VRRPv3 interoperation is optional.

2. Mixing VRRPv2 and VRRPv3 should only be done when transitioning from VRRPv2 to VRRPv3. Mixing the two versions should not be considered a permanent solution.

13.4.2. VRRPv3 Support of VRRPv2

As mentioned above, this support is intended for upgrade scenarios and is NOT RECOMMENDED for permanent deployments.

An implementation MAY implement a configuration flag that tells it to listen for and send both VRRPv2 and VRRPv3 advertisements.

When a virtual router is configured this way and is the Active Router, it MUST send both types at the configured rate, even if sub-second.

When a virtual router is configured this way and is the Backup Router, it should time out based on the rate advertised by the Active Router. In the case of a VRRPv2 Active Router, this means it must translate the timeout value it receives (in seconds) into centiseconds. Also, a Backup Router should ignore VRRPv2 advertisements from the current Active Router if it is also receiving VRRPv3 packets from it. It MAY report when a VRRPv3 Active Router is not sending VRRPv2 packets as this suggests they don’t agree on whether they’re supporting VRRPv2 interoperation.

13.4.3. VRRPv3 Support of VRRPv2 Considerations

13.4.3.1. Slow, High-Priority Active Routers

See also Section 11.2.7, "Maximum Advertisement Interval (Max Adver Int)".

The VRRPv2 Active Router interacting with a sub-second VRRPv3 Backup router is the most important example of this.

A VRRPv2 implementation should not be given a higher priority than a VRRPv2/VRRPv3 implementation it is interoperating with a VRRPv2/ VRRPv3 router's advertisement rate is sub-second.

13.4.3.2. Overwhelming VRRPv2 Backups

It seems possible that a VRRPv3 Active Router sending at centisecond rates could potentially overwhelm a VRRPv2 Backup Router with potentially non-deterministic results.

In this upgrade case, a deployment should initially run the VRRPv3 Active Routers with lower frequencies, e.g., 100 centiseconds, until the VRRPv2 routers are upgraded. Then, once the deployment has verified that VRRPv3 is working properly, the VRRPv2 support may be disabled and then the desired sub-second rates may configured.

14. Security Considerations

VRRP for IPvX does not currently include any type of authentication. Earlier versions of the VRRP specification included several types of authentication ranging from none to strong. Operational experience and further analysis determined that these did not provide sufficient security to overcome the vulnerability of misconfigured secrets, causing multiple Active Routers to be elected. Due to the nature of the VRRP protocol, even if VRRP messages are cryptographically protected, it does not prevent hostile nodes from behaving as if they are a VRRP Active Routers, creating multiple Active Router. Authentication of VRRP messages could have prevented a hostile node from causing all properly functioning routers from going into Backup state. However, having multiple Active Routers can cause as much disruption as no routers, which authentication cannot prevent. Also, even if a hostile node could not disrupt VRRP, it can disrupt ARP and create the same effect as having all routers go into Backup state.

Some L2 switches provide the capability to filter out, for example, ARP and/or ND messages from end-hosts on a switch-port basis. This mechanism could also filter VRRP messages from switch ports associated with end-hosts and can be considered for deployments with untrusted hosts.

It should be noted that these attacks are not worse and are a subset of the attacks that any node attached to a LAN can do independently of VRRP. The kind of attacks a malicious node on a LAN can perform include:

* Promiscuously receiving packets for any router's MAC address.

* Sending packets with the router's MAC address as the source MAC address in the L2 header to tell the L2 switches to send packets addressed to the router to the malicious node instead of the router.

* Sending redirects to tell the hosts to send their traffic somewhere else.

* Sending unsolicited ND replies.

* Answering ND requests, etc.

All of these can be done independently of implementing VRRP. VRRP does not add to these vulnerabilities.

VRRP includes a mechanism (setting TTL = 255, checking on receipt) that protects against VRRP packets being injected from another remote network. This limits most vulnerabilities to attacks on the local network.

VRRP does not provide any confidentiality. Confidentiality is not necessary for the correct operation of VRRP, and there is no information in the VRRP messages that must be kept secret from other nodes on the LAN.

In the context of IPv6 operation, if SEcure Neighbor Discovery (SEND) is deployed, VRRP is compatible with the "trust anchor" and "trust anchor or CGA" modes of SEND [RFC3971]. The SEND configuration needs to give the Active and Backup Routers the same prefix delegation in the certificates so that Active and Backup Routers advertise the same set of subnet prefixes. However, the Active and Backup Routers should have their own key pairs to avoid private key sharing.

15. Contributors and Acknowledgments

The IPv6 text in this specification is based on [RFC2338]. The authors of RFC2338 are S. Knight, D. Weaver, D. Whipple, R. Hinden, D. Mitzel, P. Hunt, P. Higginson, M. Shand, and A. Lindem.

The author of [VRRP-IPv6] would also like to thank Erik Nordmark, Thomas Narten, Steve Deering, Radia Perlman, Danny Mitzel, Mukesh Gupta, Don Provan, Mark Hollinger, John Cruz, and Melissa Johnson for their helpful suggestions.

The IPv4 text in this specification is based on [RFC3768]. The authors of that specification would like to thank Glen Zorn, Michael Lane, Clark Bremer, Hal Peterson, Tony Li, Barbara Denny, Joel

Halpern, Steve Bellovin, Thomas Narten, Rob Montgomery, Rob Coltun, Radia Perlman, Russ Housley, Harald Alvestrand, Steve Bellovin, Ned Freed, Ted Hardie, Russ Housley, Bert Wijnen, Bill Fenner, and Alex Zinin for their comments and suggestions.

Thanks to Steve Nadas for his work merging/editting [RFC3768] and [VRRP-IPv6] into the draft that eventually became RFC 5798 [RFC5798].

Thanks to Stewart Bryant, Sasha Vainshtein, and Pascal Thubert for comments on the current document (RFC 5798 BIS). Thanks to Gyan Mishra, Paul Congdon, and Jon Rosen for discussions related to the removal of legacy technology appendicies.

16. IANA Considerations

IANA has assigned the IPv4 multicast address 224.0.0.18 for VRRP.

IANA has assigned an IPv6 link-local scope multicast address for VRRP for IPv6. The IPv6 multicast address is FF02:0:0:0:0:0:0:12.

IANA has reserved a block of IANA Ethernet unicast addresses for VRRP for IPv6 in the range 00-00-5E-00-02-00 to 00-00-5E-00-02-FF (in hex).

17. 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>.

[RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, DOI 10.17487/RFC2460, December 1998, <https://www.rfc-editor.org/info/rfc2460>.

[RFC3768] Hinden, R., Ed., "Virtual Router Redundancy Protocol (VRRP)", RFC 3768, DOI 10.17487/RFC3768, April 2004, <https://www.rfc-editor.org/info/rfc3768>.

[RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, DOI 10.17487/RFC4291, February 2006, <https://www.rfc-editor.org/info/rfc4291>.

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

[RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, DOI 10.17487/RFC4861, September 2007, <https://www.rfc-editor.org/info/rfc4861>.

[RFC5798] Nadas, S., Ed., "Virtual Router Redundancy Protocol (VRRP) Version 3 for IPv4 and IPv6", RFC 5798, DOI 10.17487/RFC5798, March 2010, <https://www.rfc-editor.org/info/rfc5798>.

[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>.

18. Informative References

[IPSTB] Higginson, P. and M. Shand, "Development of Router Clusters to Provide Fast Failover in IP Networks", Digital Technical Journal, Volume 9 Number 3", 1997.

[NISTIR8366] "Guidance for NIST Staff on Using Inclusive Language in Documentary Standards, National Institute of Standards and Technology (NIST) Interagency or Internal Report 8366", NISTIR 8366, April 2021, <https://doi.org/10.6028/NIST.IR.8366>.

[RFC1071] Braden, R., Borman, D., and C. Partridge, "Computing the Internet checksum", RFC 1071, DOI 10.17487/RFC1071, September 1988, <https://www.rfc-editor.org/info/rfc1071>.

[RFC1256] Deering, S., Ed., "ICMP Router Discovery Messages", RFC 1256, DOI 10.17487/RFC1256, September 1991, <https://www.rfc-editor.org/info/rfc1256>.

[RFC2131] Droms, R., "Dynamic Host Configuration Protocol", RFC 2131, DOI 10.17487/RFC2131, March 1997, <https://www.rfc-editor.org/info/rfc2131>.

[RFC2281] Li, T., Cole, B., Morton, P., and D. Li, "Cisco Hot Standby Router Protocol (HSRP)", RFC 2281, DOI 10.17487/RFC2281, March 1998, <https://www.rfc-editor.org/info/rfc2281>.

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

[RFC2338] Knight, S., Weaver, D., Whipple, D., Hinden, R., Mitzel, D., Hunt, P., Higginson, P., Shand, M., and A. Lindem, "Virtual Router Redundancy Protocol", RFC 2338, DOI 10.17487/RFC2338, April 1998, <https://www.rfc-editor.org/info/rfc2338>.

[RFC2453] Malkin, G., "RIP Version 2", STD 56, RFC 2453, DOI 10.17487/RFC2453, November 1998, <https://www.rfc-editor.org/info/rfc2453>.

[RFC2464] Crawford, M., "Transmission of IPv6 Packets over Ethernet Networks", RFC 2464, DOI 10.17487/RFC2464, December 1998, <https://www.rfc-editor.org/info/rfc2464>.

[RFC3971] Arkko, J., Ed., Kempf, J., Zill, B., and P. Nikander, "SEcure Neighbor Discovery (SEND)", RFC 3971, DOI 10.17487/RFC3971, March 2005, <https://www.rfc-editor.org/info/rfc3971>.

[RFC9131] Linkova, J., "Gratuitous Neighbor Discovery: Creating Neighbor Cache Entries on First-Hop Routers", RFC 9131, DOI 10.17487/RFC9131, October 2021, <https://www.rfc-editor.org/info/rfc9131>.

[VRRP-IPv6] Hinden, R. and J. Cruz, "Virtual Router Redundancy Protocol for IPv6", Work in Progress, March 2007.

Authors' Addresses

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

Aditya Dogra
Cisco Systems
Sarjapur Outer Ring Road
Bangalore 560103
Karnataka
India
Email: addogra@cisco.com

IRTF
Internet-Draft
Intended status: Informational
Expires: 2 December 2022

M. Boucadair
Orange
D. Trossen
Huawei
A. Farrel
Old Dog Consulting

31 May 2022

Considerations for the use of SDN in Semantic Routing Networks
draft-boucadair-irtf-sdn-and-semantic-routing-01

Abstract

The forwarding of packets in today's networks has long evolved beyond ensuring mere reachability of the receiving endpoint. Instead, other 'purposes' of communication, e.g., ensuring quality of service of delivery, ensuring protection against path failures through utilizing more than one, and others, are realized by many extensions to the original reachability purpose of IP routing.

Semantic Routing defines an approach to realizing such extended purposes beyond reachability by instead making routing and forwarding decisions based, not only on the destination IP address, but on other information carried in an IP packet. The intent is to facilitate enhanced routing decisions based on this information in order to provide differentiated forwarding paths for specific packet flows.

Software Defined Networking (SDN) places control of network elements (including all or some of their forwarding decisions) within external software components called controllers and orchestrators. This approach differs from conventional approaches that solely rely upon distributed routing protocols for the delivery of advanced connectivity services. By doing so, SDN aims to enable network elements to be simplified while still performing forwarding function.

This document examines the applicability of SDN techniques to Semantic Routing and provides considerations for the development of Semantic Routing solutions in the context of SDN.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

Service differentiation in the network can be enforced by manipulating a set of parameters that belong to distinct dimensions (e.g., forwarding, routing, traffic classification, resource partitioning). Through this, the resulting system may be able to realize communication that goes beyond the mere reachability that original IP routing (and forwarding) aimed at. As pointed out in [I-D.trossen-rtgwg-routing-beyond-reachability], this differentiation and its solutions have long found entry into many existing and deployed Internet technologies.

Among the techniques to achieve such differentiation, this document focuses on Semantic Routing, which refers to a process that is meant to provide differentiated forwarding paths for specific packet flows distinct from simple shortest path first routing and, thus, satisfy specific service/application requirements.

More concretely, Semantic Routing is the process of making routing and forwarding decisions based, not only on the destination IP address of a packet, but also by taking into account other information that is carried in the packet such as (but not limited to):

* Other fields of the IP header, e.g., DSCP/Traffic Class.

* The transport header, e.g., transport port numbers [RFC7597] or subflows [RFC8803].

* Specific transport encapsulation shims, e.g., [RFC8926].

* Specific service headers, e.g., [RFC8300].

* Metadata.

Section 3 provides more details about Semantic Routing.

Software Defined Networking (SDN) places (partial or full) control of network elements and their forwarding decisions within dedicated software components called controllers and orchestrators. This approach differs from those that solely rely upon distributed routing protocols. An ambition of SDN is to enable network elements to be simplified while the network is optimized to deliver value-added connectivity services. Refer to Section 2 for an overview of SDN.

This document examines the applicability of SDN to Semantic Routing though programbale forwarding (see Section 4 and provides considerations for the development of Semantic Routing solutions in the context of SDN.

This document does not elaborate on specific SDN protocols: some SDN protocol solutions may be more or less amenable to use for Semantic Routing, but that discussion would need detailed analysis which is better suited to a further and separate document.

2. Software Defined Networking (SDN): An Overview

SDN refers to an approach for network programmability: the capacity to initialize, control, and manage network behavior dynamically via open interfaces. Such programmability can facilitate the delivery of services in a deterministic, dynamic, and scalable manner.

SDN emphasizes the role of software in operational networks by supporting the separation between data and control planes. Even if such a separation has been adopted by most routing processes for decades (Section 2.1 of [RFC7149]), SDN focuses more on the power of "central" controllers to optimize route computation within a network before populating the Forwarding Information Base (FIB) of the network elements.

The separation of the control and data planes allows faster innovation in both planes, and it enables a dynamic and flexible approach to implementing new network behaviors as well as to reacting to changes in network state and traffic demands.

SDN has been discussed in many places during the last decade. For example, within the IRTF, [RFC7426] provides a concise reference for the SDN research community to address the questions of what SDN is, what the layer structure of an SDN architecture is, and how layers interface with each other within that architecture. [RFC7149] (published in the IETF stream) offers a service provider’s perspective of the SDN landscape by describing requirements, issues, and other considerations about SDN. In particular, [RFC7149] classifies SDN techniques into the following functional domains:

* Techniques for the dynamic discovery of network topology, devices, and capabilities, along with relevant information and data models that are meant to precisely document such topology, devices, and their capabilities.

* Techniques for exposing network services and their characteristics and for dynamically capturing the set of service parameters that will be used to measure the level of quality associated with the delivery of a given service or a combination thereof.

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

* Dynamic feedback mechanisms that are meant to assess how efficiently a set of policies are enforced from a service fulfillment and assurance perspective.

SDN can be deployed in a recursive model that involves dedicated interfaces for both network and service optimization. Indeed, [RFC8597] differentiates the control functions associated with transport (that is, the transfer capabilities offered by a networking infrastructure) from those related to services in an approach called Cooperating Layered Architecture for Software-Defined Networking (CLAS).

To an SDN context, domain-specific controllers can be deployed with specific interactions as discussed in Section 4 of [RFC8309].

3. Semantic Routing: Summary of Required Technical Elements

As described in [I-D.farrel-irtf-introduction-to-semantic-routing], Semantic Routing (or, more generally, Semantic Networking) is the process of achieving enhanced routing and forwarding decisions based on semantics added to IP packet headers to provide differentiated paths for different packet flows distinct from simple shortest path first routing. The additional information or "semantics" may be placed in existing header fields (such as the IPv6 Traffic Class field or the destination address) or may be carried by adding fields to the header. Further, the semantics may be encoded in the payload or additional headers (such as in the port number fields or in an IPv6 Extension Header).

The application of Semantic Routing allows packets from different flows (even those between the same applications on the same devices) to be marked for different treatment in the network. The packets may then be routed onto different paths according to the capabilities and states of the network links in order to meet the requirements of the flows. For example, one flow may need low latency, while another may require ultra low jitter, and a third may demand very high bandwidth.

Three elements are needed to achieve Semantic Routing:

* The capabilities and state of the network must be discovered.

* The packets must be marked (with semantic information) according to their required delivery characteristics.

* The routers must be programmed to forward the traffic according to how the packets are marked.

All these elements can be matched to the SDN functional domains listed in Section 2. From that standpoint, this document provides more details on how SDN can be used to satisfy specific Semantic Routing needs.

4. Programmable Forwarding

Programmable Forwarding is the term applied to the use of control techniques to instruct network devices how to forward packets in a programmatic way.

4.1. Motivation

Modern networks are designed to carry traffic that belongs to a variety of services/applications that have distinct traffic performance requirements, reliability and robustness expectations, and service-specific needs [RFC7665][RFC8517]. Such expectations, and other forwarding requirements that can be captured in a Service Level Agreement (SLA) [RFC7297], can be considered by providers when designing their networks in order to be able to deliver differentiated forwarding behaviors. However, conventional routing and forwarding procedures do not always offer the required functionalities for such differentiated service delivery. Thus, additional means have to be enabled in these networks for the sake of innovative service delivery while minimizing the induced complexity to operate such networks. Also, these means should be tweaked to ensure consistent forwarding behaviors network-wide.

The aforementioned means are not only extensions to routing protocols, but include other mechanisms that affect the forwarding behaviors within a network. A non-exhaustive list of sample capabilities that can be offered by appropriate control of forwarding elements is provided below:

Resource Pooling: A network may host dedicated functions that implement resource pooling among many available paths or that control which path is used to steer traffic as a function of the observed round-trip time (RTT) (e.g., enable Mutlipath TCP (MPTCP) converters [RFC8803] in specific network segments, including data centers as detailed in Section 2.1 of [RFC8041]).

There is a need to interact with the underlying forwarding elements to communicate a set forwarding policies that will ensure that such service differentiation is provided to the specific flows. These forwarding policies include, for example, a set of rules that characterize the flows that are eligible to the resource pooling service or the scheduling policies (maximize link utilization, grab extra resources only when needed, etc.).

These polices are then enforced by programmable forwarders.

Performance-based Route Selection: Some applications may have strict traffic performance requirements (e.g., a low one-way delay [RFC7679]), however, the underlying network elements might not support a mechanism to disseminate performance metrics associated with specific paths and/or perform performance-based route selection (e.g., [I-D.ietf-idr-performance-routing]).

As an alternative, an off-line Semantic Routing approach could be used to collect measurement data to reach a given content (e.g., one-way delay to reach specific data centers), perform route selection based on this data, and then program the appropriate forwarding elements accordingly.

Energy-efficient Forwarding: An important effort was made in the past to optimize the energy consumption of network elements. However, such optimization is node-specific and no standard means to optimize the energy consumption at the scale of the network have been defined. For example, many nodes (also, service cards) are deployed as backups.

A controller-based approach can be implemented so that the route selection process optimizes the overall energy consumption of a path. Such a process takes into account the current load, avoids waking nodes/cards for handling "sparse" traffic (i.e., a minor portion of the total traffic), considers node-specific data (e.g., [RFC7460]), etc. This off-line Semantic Routing approach will transition specific cards/nodes to "idle" and wake them as appropriate, etc., without breaking service objectives. Moreover, such an approach will have to maintain an up-to-date topology even if a node is in an "idle" state (such nodes may be removed from adjacency tables if they don’t participate in routing advertisements).

Network Partitioning: A network may need to be partitioned in order to rationalize the delivery of advanced connectivity services, and to address specific forwarding requirements of groups of services/ applications. Network slicing [I-D.ietf-teas-ietf-network-slices] can be considered to deliver these services. However, an

intelligence is needed to decide the criteria to be used to partition the available resources, filter them, decide whether network extensions are needed, ensure whether/how resource preemption is adequately implemented, etc.

These tasks are better achieved using a central intelligence that has direct visibility into the intents of applications, underlying network capabilities, local policies and guidelines, etc. As an output of processing these various inputs, a set of node-specific policies is generated, and then pushed using available SDN interface.

Alternative Forwarding: The programmability of SDN in the form of forwarding actions defined on packet header fields allows for realizing forwarding techniques beyond the typical longest-prefix match used for IP-based reachability. Solutions, like those in [ICC2016], use a binary representation of links in a network to realize a path-based forwarding action that acts purely on node-local state, independent of the nature of the path or the communications traversing it. As discussed in Section 7, the limitation of forwarding actions to apply only to defined (IP) packet header fields results in issues that need special consideration when realizing such solutions in real-world deployments.

The next subsection further details which elements are needed when interacting with programmable forwarders in an SDN context.

4.2. SDN for Semantic Routing: The Intended Behavior

SDN minimizes the required changes to legacy (interior) routing protocols. More concretely, SDN can be used to provide the intended Semantic Routing behavior, especially:

* Identify the forwarding elements that can be safely involved in providing the intended Semantic Routing features.

* Maintain abstract topologies that involve these elements and their capabilities.

* Capture application-specific intents and derive the corresponding forwarding requirements and, then, forwarding policies.

* Map these abstract topologies to (groups of) applications with specific Semantic Routing needs.

* Program a subset of nodes (called boundary nodes) with the required classification and marking policies to bind flows to their intended Semantic Routing behaviors.

  In order to adequately process the application flows that require specific differentiated forwarding, SDN controllers maintain a table that allows to unambiguously identify such flows. The content of that table is used to derive the appropriate classification/match rules that are then communicated by an SDN controller to a set of forwarding elements.

  When volatile data (e.g., dynamic IP addresses) are used to build such rules, it is the responsibility of the SDN controllers to update the rules whenever a new identifier is used. Failure to maintain "fresh" classification rules will lead to service failure/degradation.

* Supply intermediate nodes (that is, nodes that are not boundary nodes) with the appropriate rules to locate and interpret the bits within the packet to determine and execute forwarding actions as established by Semantic Routing.

* Automatically adjust, if possible, the network MTU to accommodate any overhead that is introcuced by any extra bits used to signal Semantic Routing behavior.

* Instruct egress boundary nodes about the required actions such as stripping or setting any Semantic Routing bits.

* Interact with the underlying nodes to maintain, retrieve, and disseminate the data that are used for assuring that Semantic Routing policies are appropriately fulfilled.

* Configure OAM policies to measure the network behavior and adjust the forwarding processes.

* Monitor the network and detect parts of the network where policies are broken or suboptimal.

* Automate the overall procedure [RFC8969].

At least three approaches can be considered by an SDN controller to accomplish the above tasks:

* Compute (centrally) the differentiated paths and install the required forwarding rules in involved nodes. Strict or loose paths may be installed. This approach has the merit of implementing new path selection algorithms without requiring them to be supported by every involved node.

* Assign (centrally) differentiated link information and install the required forwarding rules in the involved nodes. End-to-end paths are constructed without involvement of the SDN controller, utilizing the link information to establish path identifiers on which installed forwarding rules can act upon without additional path-specific knowledge being required. See [ICC2016] for an example of such an approach.

* Rely upon a distributed routing protocol to customize the route selection process ([I-D.ietf-lsr-flex-algo], for example). In such cases, the SDN controller is responsible for communicating the parameters to be used for the route selection process, selecting the nodes that will participate in a given topology, and configuring any tunnels to interconnect these nodes.

A hierarchical SDN design can also be considered, where specific controllers are enabled in each domain with dedicated interfaces to share data (e.g., radio bottlenecks, expectations). These domains do not need to support the same technological implementations. The interaction between the SDN controllers eases the delivery of consistent Semantic Routing behaviors without requiring common domain configuration.

5. Policy-Based Semantic Routing

Policy is a term applied to the application of local or network-wide operational choices made by the network manager. These may range from decisions about what traffic to admit to the network, how network resources should be used to support different traffic flows, how errors or security violations are handled, and how packets are routed through the network.

Policies are usually made available to network operators as configuration elements on network nodes. However, these configuration actions need to be coordinated across the whole network if the policies are to be effective. Thus, a mechanism is desired that allows an operator to set a network-wide policy in one place and that results in that policy being pushed out to the network nodes that need to act on the policy.

Semantic Routing is particularly amenable to a policy-based approach. That is, an operator (or their software tools) can make decisions about how different traffic flows should be handeled in the network. Those decisions can then be installed on network nodes so that different traffic is handled differently and according to the policies.

SDN is a powerly approach to implement a policy-based network management framework. The operator need only select or configure the desired policies at the controller: the controller will realize the policies and install the necessary instructions and behaviors on the network nodes.

6. Network-Wide Coordination

Critical to the correct functioning of any routing system is proper network-wide coordination. In many cases, the coordination starts with the collection and dissemination of network connectivity information (known as the network topology), the capabilities of the network nodes and links, and the current state (up, down, degraded, busy, etc.) of those nodes and links. But an even mode fundamental element of network-wide coordination is the decision about which routing algorithms and procedures will be used because, if different nodes or even different parts of the network) apply different routing approaches, it is very possible that traffic will loop or be dropped. Thus, th first elements of coordination are finding out what the network looks like and agreeing how to route traffic.

These essentials are no less relevant in Semantic Routing. All nodes that participate in a Semantic Routing network need to have the same understanding of the additional information carried in packets, and must make coordinated forwarding decisions based on a coordinated routing algorithm.

A centralized approach, such as that achieved in an SDN system, is particularly useful in this context because it allows the coordination to be applied through a central point of control which may remove the complexity and "fragility" from the routing system. This coordination may be considered in parallel with the aspects of policy-based routing described in Section 5.

7. Applying Semantic Information to Packets

Given the focus of Semantic Routing is the use within IP networks, semantic information that can be used in SDN-based Semantic Routing is limited to those fields specifically defined for use with Semantic Routing (see Section 2 for more information). This document deliberately makes no comment on the specifications that may be produced to define such fields, their meaning, and their encoding.

SDN aligns with the concept of Semantic Routing in that it allows the network devices to be programmed for forwarding actions indicated by a wide range of packet header fields beyond simply the IP destination addresses.

However, Semantic Routing solutions have also been proposed that "overwrite" existing protocol fields in order for them to carry semantic information that can be used to drive a forwarding action outside their original semantics. [POINT2015] and [POINT2016] outline an example of such approaches in which semantic information is used for a path-based forwarding decision; while the absence of "path" information is foreseen as an actionable packet header field in IPv6.

Here, the path is constructed by a Path Computation Element (PCE) [RFC4655] that matches a given service name against previously announced locations where said service name is located. The path is represented as a concatenation of individual link information, which in pushed by the SDN controller to the network nodes so that they can perform local forwarding actions on packets that arrive. Given the binary structure of the end-to-end path information, the forwarding operation can be implemented in a standard-compliant manner with its realization described in [ICC2016] as an arbitrary wildcard matching operation.

However, the constraint of acting only on limited packet fields requires that the path information be carried in one of those standard-defined packet header fields: thereby overwriting (or overloading) any existing packet header field. [POINT2016] uses the IPv6 address fields for this purpose, representing the longest continuous binary field in the IPv6 header (two addresses make up 256 bits in total) allows the support of topologies with up to 256 links.

Given the approach chosen in [POINT2016], any IPv6 address information, if needed, cannot be present in the packet header and so is provided in the encapsulated payload. This leads to repeated encapsulation with the overhead of carrying two IP headers in a single packet: one used for path-based forwarding and one for the operations in arriving endpoint. Only newer SDN-based forwarding

plane programming tools, such as P4, would allow for such overhead to be removed by placing the path information into another packet header field (or even the payload as an extended header of sort) to act upon.

8. Benefits and Concerns with the Use of SDN for Semantic Routing

The programmability of SDN provides a fertile ground for forwarding decision that go beyond the reachability information provided through IPv4/v6 addresses, e.g., by using other packet header fields. This not only allows for extending the simple reachability-driven forwarding decision with richer, e.g., policy-based, decisions (as discussed in Section 5), it may also enable new forwarding paradigms per se, such as those in [POINT2016], which in turn may realize forwarding behaviours like multicast at much lower cost points and higher efficiency (see [ICC2016]).

However, SDN specifications have limited capabilities when it comes to the additional (i.e., new) packet header fields that may be used for forwarding actions. As a consequence, "true" Semantic Routing on any semantic enhancement, which is included in the packet, is only possible in a manner limited to those existing fields.

Solutions such as those in [POINT2016], using methods outlined in [ICC2016], attempt to break this limitation albeit by overwriting standard-defined packet header fields, thereby changing the semantics of those fields within the scope (i.e., network domain) where the "re-defined" semantics are known and understood.

This limits any solution to a limited domain [RFC8799]. More importantly, the redefinition of packet fields poses the danger of exposing this (non-standard compliant) semantic to elements outside the limited domain: semantic leakage may occur, or nodes outside the domain may misinterpret overwritten fields, requiring methods, such as dedicated gateways, to preventi such leakage. This can be seen in [POINT2016], where the boundaries to IP-compliant end devices and other domains alike are delimited by dedicated gateway elements. Those gateways usually act at higher layers than the forwarding layer, thereby incurring complexity and often delay.

See also [I-D.king-irtf-challenges-in-routing] for a discussion of issues and concerns that need to be examined when applying a new routing or forwarding paradigm to a self-contained network or Internet.

9. Security Considerations

SDN-related considerations are discussed in Section 5 of [RFC7149].

10. IANA Considerations

This document makes no requests for IANA action.

11. Acknowledgements

Thanks to George Polyzos for helpful comments on this work.

This work is partially supported by the European Commission under Horizon 2020 grant agreement number 101015857 Secured autonomic traffic management for a Tera of SDN flows (Teraflow).

12. Contributors

George Xylomenos
Email: xgeorge@aueb.gr

13. Informative References

[I-D.farrel-irtf-introduction-to-semantic-routing]
Farrel, A. and D. King, "An Introduction to Semantic Routing", Work in Progress, Internet-Draft, draft-farrel-irtf-introduction-to-semantic-routing-04, 25 April 2022, <https://www.ietf.org/archive/id/draft-farrel-irtf-introduction-to-semantic-routing-04.txt>.

[I-D.ietf-idr-performance-routing]
Xu, X., Hegde, S., Talaulikar, K., Boucadair, M., and C. Jacquenet, "Performance-based BGP Routing Mechanism", Work in Progress, Internet-Draft, draft-ietf-idr-performance-routing-03, 22 December 2020, <https://www.ietf.org/archive/id/draft-ietf-idr-performance-routing-03.txt>.

[I-D.ietf-lsr-flex-algo]
Psenak, P., Hegde, S., Filsfils, C., Talaulikar, K., and A. Gulko, "IGP Flexible Algorithm", Work in Progress, Internet-Draft, draft-ietf-lsr-flex-algo-20, 18 May 2022, <https://www.ietf.org/archive/id/draft-ietf-lsr-flex-algo-20.txt>.

[I-D.ietf-teas-ietf-network-slices]
Farrel, A., Drake, J., Rokui, R., Homma, S., Makhijani, K., Contreras, L. M., and J. Tantsura, "Framework for IETF

Network Slices", Work in Progress, Internet-Draft, draft-ietf-teas-ietf-network-slices-10, 27 March 2022, <https://www.ietf.org/archive/id/draft-ietf-teas-ietf-network-slices-10.txt>.

[I-D.king-irtf-challenges-in-routing]
King, D., Farrel, A., and C. Jacquenet, "Challenges for the Internet Routing Systems Introduced by Semantic Routing", Work in Progress, Internet-Draft, draft-king-irtf-challenges-in-routing-08, 25 April 2022, <https://www.ietf.org/archive/id/draft-king-irtf-challenges-in-routing-08.txt>.

[I-D.trossen-rtgwg-routing-beyond-reachability]
Trossen, D., Lou, D., and S. Jiang, "Continuing to Evolve Internet Routing Beyond 'Mere' Reachability", Work in Progress, Internet-Draft, draft-trossen-rtgwg-routing-beyond-reachability-00, 8 February 2022, <https://www.ietf.org/archive/id/draft-trossen-rtgwg-routing-beyond-reachability-00.txt>.

[ICC2016] Reed, M., Al-Naday, M., Thomos, N., Trossen, D., Petropoulos, G., and S. Spirou, "Stateless multicast switching in software defined networks", Paper IEEE ICC 2016, 2016.

[POINT2015]
Trossen, D., Reed, M., Riihijarvi, J., Georgiades, M., Xylomenos, G., and S. Fotiou, "IP Over ICN: The Better IP?", Paper EuCNC (European Conference on Networks and Communications), Paris, France, 2015.

[POINT2016]
Kim, S.-Y.., Robitzsch, S., Trossen, D., Reed, M., Al-Naday, M., and J. Riihijarvi, "Realizing IP-based Services over an Information-Centric Networking Transport Network", Paper Proceedings of the 3rd ACM Conference on Information-Centric Networking, Pages 215-216, 2016.

[RFC4655] Farrel, A., Vasseur, J.-P., and J. Ash, "A Path Computation Element (PCE)-Based Architecture", RFC 4655, DOI 10.17487/RFC4655, August 2006, <https://www.rfc-editor.org/info/rfc4655>.

[RFC7149] Boucadair, M. and C. Jacquenet, "Software-Defined Networking: A Perspective from within a Service Provider Environment", RFC 7149, DOI 10.17487/RFC7149, March 2014, <https://www.rfc-editor.org/info/rfc7149>.

[RFC7297] Boucadair, M., Jacquenet, C., and N. Wang, "IP Connectivity Provisioning Profile (CPP)", RFC 7297, DOI 10.17487/RFC7297, July 2014, <https://www.rfc-editor.org/info/rfc7297>.

[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>.

[RFC7460] Chandramouli, M., Claise, B., Schoening, B., Quittek, J., and T. Dietz, "Monitoring and Control MIB for Power and Energy", RFC 7460, DOI 10.17487/RFC7460, March 2015, <https://www.rfc-editor.org/info/rfc7460>.

[RFC7597] Troan, O., Ed., Dec, W., Li, X., Bao, C., Matsushima, S., Murakami, T., and T. Taylor, Ed., "Mapping of Address and Port with Encapsulation (MAP-E)", RFC 7597, DOI 10.17487/RFC7597, July 2015, <https://www.rfc-editor.org/info/rfc7597>.

[RFC7665] Halpern, J., Ed. and C. Pignataro, Ed., "Service Function Chaining (SFC) Architecture", RFC 7665, DOI 10.17487/RFC7665, October 2015, <https://www.rfc-editor.org/info/rfc7665>.

[RFC7679] Almes, G., Kalidindi, S., Zekauskas, M., and A. Morton, Ed., "A One-Way Delay Metric for IP Performance Metrics (IPPM)", STD 81, RFC 7679, DOI 10.17487/RFC7679, January 2016, <https://www.rfc-editor.org/info/rfc7679>.

[RFC8041] Bonaventure, O., Paasch, C., and G. Detal, "Use Cases and Operational Experience with Multipath TCP", RFC 8041, DOI 10.17487/RFC8041, January 2017, <https://www.rfc-editor.org/info/rfc8041>.

[RFC8300] Quinn, P., Ed., Elzur, U., Ed., and C. Pignataro, Ed., "Network Service Header (NSH)", RFC 8300, DOI 10.17487/RFC8300, January 2018, <https://www.rfc-editor.org/info/rfc8300>.

[RFC8309] Wu, Q., Liu, W., and A. Farrel, "Service Models Explained", RFC 8309, DOI 10.17487/RFC8309, January 2018, <https://www.rfc-editor.org/info/rfc8309>.

[RFC8517] Dolson, D., Ed., Snellman, J., Boucadair, M., Ed., and C. Jacquenet, "An Inventory of Transport-Centric Functions Provided by Middleboxes: An Operator Perspective", RFC 8517, DOI 10.17487/RFC8517, February 2019, <https://www.rfc-editor.org/info/rfc8517>.

[RFC8597] Contreras, LM., Bernardos, CJ., Lopez, D., Boucadair, M., and P. Iovanna, "Cooperating Layered Architecture for Software-Defined Networking (CLAS)", RFC 8597, DOI 10.17487/RFC8597, May 2019, <https://www.rfc-editor.org/info/rfc8597>.

[RFC8799] Carpenter, B. and B. Liu, "Limited Domains and Internet Protocols", RFC 8799, DOI 10.17487/RFC8799, July 2020, <https://www.rfc-editor.org/info/rfc8799>.

[RFC8803] Bonaventure, O., Ed., Boucadair, M., Ed., Gundavelli, S., Seo, S., and B. Hesmans, "0-RTT TCP Convert Protocol", RFC 8803, DOI 10.17487/RFC8803, July 2020, <https://www.rfc-editor.org/info/rfc8803>.

[RFC8926] Gross, J., Ed., Ganga, I., Ed., and T. Sridhar, Ed., "Geneve: Generic Network Virtualization Encapsulation", RFC 8926, DOI 10.17487/RFC8926, November 2020, <https://www.rfc-editor.org/info/rfc8926>.

[RFC8969] Wu, Q., Ed., Boucadair, M., Ed., Lopez, D., Xie, C., and L. Geng, "A Framework for Automating Service and Network Management with YANG", RFC 8969, DOI 10.17487/RFC8969, January 2021, <https://www.rfc-editor.org/info/rfc8969>.

Authors' Addresses

Mohamed Boucadair
Orange
Rennes
France
Email: mohamed.boucadair@orange.com

Dirk Trossen
Huawei
Munich
Germany
Email: dirk.trossen@huawei.com

Adrian Farrel
Old Dog Consulting
United Kingdom
Email: adrian@olddog.co.uk

IRTF
Internet-Draft
Intended status: Informational
Expires: 27 October 2022

A. Farrel
Old Dog Consulting
D. King
Lancaster University

25 April 2022

An Introduction to Semantic Routing
draft-farrel-irtf-introduction-to-semantic-routing-04

Abstract

Many proposals have been made to add semantics to IP packets by placing additional information in existing fields, by adding semantics to IP addresses themselves, or by adding fields. The intent is to facilitate enhanced routing/forwarding decisions based on these additional semantics to provide differentiated forwarding paths for different packet flows distinct from simple shortest path first routing. The process is defined as Semantic Routing.

This document provides a brief introduction to Semantic Routing.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

Historically, the meaning of an IP address has been to identify an interface on a network device or a network to which a host is attached [RFC0814]. Network routing protocols were initially designed to determine paths through a network toward destination addresses so that IP packets with a common destination address converged on that destination. Anycast and multicast addresses were also defined (e.g., Section 2.6.1 of [RFC4291]), and some of these new address semantics necessitated variations to the routing protocols (e.g., [RFC6992]), and in some cases the development of new routing protocols (e.g., Protocol Independent Multicast - Sparse Mode [RFC7761]).

Over time, routing decisions were enhanced to route packets according to additional information carried within the packets and dependent on policy coded in, configured at, or signaled to the routers. Perhaps the most obvious example is Equal-Cost Multipath (ECMP) where a router makes a consistent choice for forwarding packets over a number of parallel links or paths based on the values of a set of fields in the packet header. Another example is Constraint-based Shortest Path First (CSPF) where additional constraints are considered when performing route computation and selection.

Upper-layer applications are placing increasingly sophisticated demands on the network for better quality, more predictability, and increased reliability. Some of these applications are futuristic predictions (for example, haptic augmented reality multiplayer 3D worlds), some are new ideas on the threshold of roll-out (such as holographic conferencing), and many are rapidly developing sectors with established revenue streams (such as multiplayer immersive gaming).

At the same time, lower-layer network technologies are advancing rapidly providing increased bandwidth to the home and to mobile hand-held devices. These advances create an environment that enables the potential of advanced applications being run by very many end-users. This coincides with a massive growth in end-to-end communications that include machines and services, and to introduce routing and addressing behaviors to a particular use case and set of requirements applied within a limited region or domain of the Internet. Examples of these three developments include 5G, predicted wireless evolutions, IoT and vehicular connectivity, space-terrestrial communication, industrial networks, cloud computing, service function chaining and network functions virtualization, digital twins, and data-centric data brokerage platforms.

Despite this plurality of communication scenarios, IP-based addressing and network layer routing have remained focused on identifying locations of communication (i.e., "where") and determining paths between those locations with or without specific constraints (i.e., "how-to-get-there" as per [IEN23]). This has previously depended on higher-layer capabilities (e.g., for name-to-location resolution) to support some of these communication scenarios, but that approach introduces latency and dependencies (e.g., changing locator assignments may depend on the capabilities of the upper-layer capability that are outside the core addressing and routing system). Furthermore, multi-layer lookups and interactions may impact the efficacy of some of the communication scenarios mentioned here, particularly those that employ different routing and addressing approaches beyond just locators.

"Semantic Routing" places the support for advanced routing, forwarding, and location functions directly at the packet routing/forwarding layer, such as through extensions to the identification properties of addresses (so that the address indicates more than just the network location) or through performing routing functions on an extended set of inputs (for example, other fields carried in packet headers). Such an approach should preserve the Internet architecture as it is today while enabling additional routing function.

This document provides a brief introduction to semantic routing and outlines the possible approaches that might be taken. A separate document ([I-D.king-irtf-semantic-routing-survey]) makes a start at a survey of pre-existing work in this area, while [I-D.king-irtf-challenges-in-routing] sets out some of the issues that should be considered when researching, developing, or proposing a semantic routing scheme.

2. Objectives and Scope

As with all advances in Internet protocols, semantic routing may be considered for Internet-wide deployment or may be restricted (possibly only initially) to well-defined and contained networks referred to as "limited domains" (see [RFC8799]). The information used for semantic routing may be opaque within the network (in other words, the additional information is not required to be parsed by the routers and might not even be visible to them), may be transparent (so that routers may see the information, but their processing does not need to be changed to accommodate the information or its encoding), or may be active (so that semantic routing is fully enabled).

When building an end-to-end path across multiple domains, semantic routing may select a path in one domain that is not consistent with the paths selected in other domains in terms of constructing the "best" end-to-end path. That is, the semantic routing decisions within a domain are potentially isolated from knowledge about the other domains.

In any case, concern and consideration must be coexistence with, and backward compatibility to, existing routing and addressing schemes that are widely deployed.

Further understanding of the scope of semantic routing applied to the routing of packets at the network layer may be gained by reading Section 6 to see how various other concepts of routing are out of scope of this work.

A strategic objective of semantic routing, and associated semantic enhancements, is to enable Service Providers to modify the default forwarding behaviour to be based on other information present in the packet and policy configured or dynamically programmed into the routers and devices. This is aimed to cause new and alternative path processing by routers, including:

* Determinism of quality of delivery in terms of throughput, latency, jitter, and drop precedence.

* Determinism of resilience in terms of survival of network failures and delivery degradation.

* Determinism of routing performance in terms of the volume of data that has to be exchanged both to establish and to maintain the routing tables.

* Deployability in terms of configuration, training, development of new hardware/software, and interaction with the pre-existing network technologies and uses.

* Efficiency of manageability in terms of:

  1. diagnostic management

  2. management of Service KPIs with/without guarantees

  3. dynamic and controlled instantiation of management information in the packets.

Issues of security and privacy have been largely overlooked within the routing systems. However, there is increasing concern that attacks on routing systems can not only be disruptive (for example, causing traffic to be dropped), but may cause traffic to be routed via inspection points that can breach the security or privacy of the payloads (e.g., BGP hijack attacks). While semantic routing might offer tools for increasing security and privacy, it is possible that semantic routing and the additional information that may be carried in packets to enable semantic routing may provide vectors for attacks or compromise privacy. This must be examined by any semantic routing proposals. For example, means to control entities that are entitled to access supplied semantic routing information should be considered.

3. Approaches to Semantic Routing

Typically, in an IP-based network packets are forwarded using the least-cost path to the destination IP address. Service Providers may also use techniques to modify the default forwarding behavior based on other information present in the packet and configured or programmed into the routers. These mechanisms, sometimes called semantic routing techniques include "Preferential Routing", "Policy-based Routing", and "Flow Steering".

Examples of existing semantic routing usage in IP-based networks include the following.

* Using addresses to identify different device types so that their traffic may be handled differently [SEMANTICRTG].

* Expressing how a packet should be handled, prioritized, or allocated network resources as it is forwarded through the network [TERASTREAMref].

* Deriving IP addresses from the lower layer identifiers and using addresses depending on the underlying connectivity (for example, [RFC6282].

* Building IP addresses from the transport layer identifiers (for example, [RFC7597]).

* Indicating the application or network function on a destination device or at a specific location, or enable Service Function Chaining (SFC) [RFC7665].

* Providing semantics specific to mobile networks so that a user or device may move through the network without disruption to their service [CONTENT-RTG-MOBILEref].

* Enabling optimized multicast traffic distribution by encoding multicast tree and replication instructions within addresses [MULTICAST-SRref].

* Content-based routing (CBR), forwarding of the packet based on message content rather than the destination addresses [OPENSRNref].

* Identifying hierarchical connectivity so that routing can be simplified [EIBPref].

* Providing geographic location information within addresses [GEO-IPref].

* Using cryptographic algorithms to mask the identity of the source or destination, masking routing tables within the domain, while still enabling packet forwarding across the network [BLIND-FORWARDINGref].

A more comprehensive list of existing implementations and research projects can be found in [I-D.king-irtf-semantic-routing-survey].

Semantic routing, operates to forward packets dependent on information carried in the packets and rules present in the routers. Those rules could be any combination of:

* Built into the routers

* Configured network-wide in the routers

* Configured per-router in a relatively static way

* Programmed to the routers in a dynamic way, for example, through software defined networking (SDN)

* Distributed dynamically through the network using routing or signalling protocols

Semantic routing will also require information about network state and capabilities just as existing shortest path first routing systems do. That may require information (such as link delays or other qualitative attributes) to be collected by network nodes and distributed between routers by routing protocols. Alternatively, this information could be collected (centrally) by a set of network controllers and used to derive the rules installed in the routers.

Forwarding by a router is based on a look-up that also considers the semantic routing information carried in the packet (see Section 4) and forwarding instructions programmed into the forwarding element. Some semantic routing proposals may generate the semantic information (e.g., a hash) rather than using information that is directly extracted from the packet. The actions to perform may be derived by the router based on the rules and information that the router has collected, or may be programmed directly from the network controller.

3.1. Packet and Service Routing

Routing is the process of selecting a path for traffic in a network or between or across multiple networks. For example, IP routing uses IP addresses for source and destination identification and is typically used for packet networks, such as the Internet. IP routing assumes that network addresses are structured and facilitates routing entries in a routing table entry to represent a group of IP-capable devices.

While service routing and information-centric networking (ICN) can operate directly on top of layer 2 protocols (for example, [RFC9139]), in the context of this document, we are concerned with the function of service routing and ICN in IP networks. Like any new spanning-layer style protocol, deployment considerations for ICN on the Internet make tunneling through IP a required part of any co-existence or transition. The approach taken in this case, is to create an overlay layer on top of the IP network. Control of the overlay necessitates augmentation of existing routing mechanisms, or entirely new discovery, propagation and resource management protocols and procedures.

By contrast, explicit service-based IP routing [I-D.jiang-service-oriented-ip] abstracts the service actions that the network can provide into a number of classes called Service Action Types (SATs). Each packet is marked with the relevant SAT, and the packets are routed to the next available SAT provider (not the destination IP address). In this approach, a distinct encapsulation is needed and may carry native IP packets as payload, while transition experiments may utilize an overlay on top of IP.

IP Routing and service routing are not the same thing.

4. Semantic Routing Information

The subsections below describe some of the common techniques to enable semantic routing in more detail. The sections are unordered and no meaning should be assigned to how one approach is presented before another. They are not a complete list of possible approaches.

The approaches described here have many advantages and disadvantages. The purpose here is not to determine which approach is best or most appropriate, and so those advantages and disadvantages are not discussed. The reader will inevitably have a preference and see drawbacks.

4.1. Address Space Partitioning

In some cases, an address prefix is assigned a special purpose and meaning. When such an address appears in the packet's address field, a router can know from the prefix that particular routing/forwarding actions are required. An example of this approach is seen in multicast addressing. Another example is the handling of anycast in IPv6 where the nodes to which the address is assigned must be explicitly configured to know that it is an anycast address [RFC4291].

4.2. Prefix-based Contextual Address Usage

The owner of a prefix to use the low-order bits of an address for their own purposes.

The semantics of such an approach might be coordinated between prefix owners, or could be indicated through information that is part of the encoding, and is standardized. An example of such approach is in IPv4/IPv6 Translators [RFC6052].

4.3. Semantic Addressing

Semantic addressing is a term applied to any approach that adds semantics to IP addresses. This includes the mechanisms described in Section 4.1 and Section 4.2. Other semantic addressing proposals suggest variable address lengths, hierarchical addresses, or a structure to addresses so that they can carry additional information in a common way.

In any case, semantic addressing that intends to facilitate routing decisions is based solely on the address and without the need to find and process information carried in other fields within the packets.

Note that not all semantic addressing schemes exist to facilitate routing (for example, content addressing where the interface ID of the address identifies a chunk of the content to be retrieved), but such schemes are naturally out of scope of this document.

4.4. Flow Marking

Flow marking is a way of indicating, in a specific field in the packet header, the treatment that the packet should receive in the network. In IPv4 the six-bit DSCP field is commonly used for this purpose. In IPv6, while the Traffic Class field could be used, it is generally recommended that the Flow Label field should serve this and a more general purpose.

4.5. Extended Lookup

Routers may also examine fields in the packet other than those in the IP header. For example, many router processes may look at the "five-tuple" consisting of:

* source address

* destination address

* next protocol

* transport protocol source port

* transport protocol destination port

4.6. Semantic Field Overloading

"Overloading" is a term applied to placing additional semantics on the contents of a field beyond how it is specified. This is relatively hard to do in an IPv6 header because the number of fields is small, and all fields have specific meanings that are needed in all cases. In IPv4 there may be more opportunity to use some fields in very controlled situations to carry additional semantics that can be used for semantic routing.

4.7. IPv6 Extension Headers

IPv6 defines extension headers explicitly for carrying information that may be used by routers along the path. This information can be used to instruct all routers, only the router indicated by the destination address, or by the ultimate destination of the packet.

Extension headers may carry any information to enable semantic routing.

4.8. New Extensions

Another approach is to define a new protocol extension to carry information on which semantic routing can be performed. Such an extension could be in the form of a new extension header (see Section 4.7) or as a new shim encapsulation (e.g., [RFC7665]).

5. Architectural Considerations

Some semantic routing proposals are intended to be deployed in limited domains [RFC8799] (networks) that are IP-based, while other proposals are intended for use across the Internet. The impact that the proposals have on routing systems may require clean-slate solutions, hybrid solutions, extensions to existing routing protocols, or potentially no changes at all.

Semantic data may be applied in several ways to integrate with existing routing architectures. The most obvious is to build an overlay such that IP is used only to route packets between network nodes that utilize the semantics at a higher layer. An overlay may be achieved in a higher protocol layer, or may be performed using tunneling techniques (such as IP-in-IP [RFC1853]) to traverse the areas of the IP network that cannot parse additional semantics thereby joining together those nodes that use the semantic data.

The application of semantics may also be constrained to within a limited domain. In some cases, such a domain will use IP, but be disconnected from Internet (see Section 5.1). In other cases, traffic from within the domain is exchanged with other domains that are connected together across an IP-based network using tunnels or via application gateways (see Section 5.2). And in still another case traffic from the domain is routed across the Internet to other nodes and this requires backward-compatible routing approaches (see Section 5.3).

5.1. Isolated Domains

Some IP network domains are entirely isolated from the Internet and other IP-based networks. In these cases, there is no risk to external networks from any semantic routing schemes carried out within the domain.

Many approaches in isolated domains will utilize environment-specific routing protocols. For example, those suited to constrained environments (for IoT) or mobile environments (for autonomous vehicles). Such routing protocols can be optimized for the exchange of information specific to semantic routing. However, gateways to provide external connectivity are usually deployed in such networks. Appropriate means should be supported in these means to prevent leaking semantic information beyond the boundaries of these domains.

5.2. Bridged Domains

In some deployments, it will be desirable to connect a number of isolated domains to build a larger network. These domains may be connected (or bridged) over an IP network or even over the Internet.

Ideally, the function of the bridged domains should not be impeded by how they are connected, and the operation of the IP network providing the connectivity should not be compromised by the act of carrying traffic between the domains. This can generally be achieved by tunneling the packets between domains using any tunneling technique, and this will not require the IP network to know about the semantic routing used by the domains.

An alternative to tunneling is achieved using gateway functionality where packets from a domain are mapped at the domain boundary to produce regular IP packets that are sent across the IP network to the boundary of the destination domain where they are mapped back into packets for use within that domain.

5.3. Semantic Prefix Domains

A semantic prefix domain [I-D.jiang-semantic-prefix] is a portion of the Internet over which a consistent set of semantic-based policies are administered in a coordinated fashion. This is achieved by assigning a routable address prefix (or a set of prefixes) for use with semantic addressing and routing so that packets may be routed through the regular IP network (or the Internet) using the prefix and without encountering or having to use any semantic addressing. Once delivered to the semantic prefix domain, a packet can be subjected to whatever semantic routing is enabled in the domain.

6. A Brief Discussion of What Constitutes Routing

This section provides an overview of what is considered as "routing" in the scope of this document. There are many functions in the Internet that contain the concept of routing, but not all of them apply to the scope of this document which is concerned with routing packets at the network layer. A more thorough catalogue of approaches to routing and the applications of semantic routing can be found in [I-D.king-irtf-semantic-routing-survey].

6.1. Application Layer Routing

Routing in the application layer concerns the choice of application-level components that are distributed across the network. The choice may be dependent on the services being delivered, knowledge about the locations in the network that can provide the services, knowledge of the network capabilities, and preferences expressed by an application or user. In this sense, the routing choice consists of constructing an "application layer path" and may be performed at the head end or along the path. Packets are carried between components across the underlying network, using normal transport and network layer protocols that may, themselves, involve routing. Thus, application layer routing is concerned with selecting a series of components based on the potential to carry traffic between them, but without concern for how the packets are routed within the network.

Application layer routing may be used in concepts such as Content Distribution Networking (CDN) and computation in the network (COIN) (see Section 6.9).

The ALTO architecture and protocol [RFC7285] is intended to allow the network to answer queries about the availability and characteristics of paths between application-level components to enable choices to be made by providers of function or content about which components to select. This is a server-based approach because it would be impractical to scale the network reporting all available paths to all destinations to every client, or for the network edge to be able to answer queries from their clients.

6.2. Higher-Layer Path Selection

There is another high-level path selection scenario that is more concerned with selecting outbound paths from the source than in determining destinations or next application-layer hops (as described in Section 6.1. For example, consider a mobile phone that is connected to Wi-Fi and 5G. Further, consider that the Wi-Fi network is dual-homed to two different ISPs. This gives an application a choice of three different paths depending on the known (or

advertised) capabilities of the networks.

This type of scenario is being examined by the Path Aware Networking Research Group (PANRG) where, rather than consulting a server to supply the most appropriate path, the source host or application should learn about the potential paths and pick between them.

6.3. Transport Layer Routing

Some transport layer load balancing schemes and proxy-based connection or discovery mechanisms use a mechanism that looks somewhat like routing, but exists in the transport layer. For example, section 2.1.1 of [RFC3135] describes how a transport layer Performance Enhancing Proxy (PEP) may use a concept called TCP spoofing to terminate a TCP connection and initiate a new connection to the next proxy on the transport layer path towards the destination. The IP addresses of the packets are rewritten at the proxies so that the packets can be routed/forwarded to the next proxy, but no change to the underlying routing system is implied, and this is not Semantic Routing.

6.4. Tunnel-Based Routing

Tunnel-based routing schemes, like those in the transport layer (see Section 6.3), are achieved through an overlay. a tunnel-based scheme relies on encapsulating packets so that they can be sent through the normal routing and forwarding network for delivery to an interim node. That node decapsulates the packet and then either continues to forward the contents or encapsulates the contents in another tunnel. Some approaches, such as onion routing in the Tor project (see [ONION]) use a scheme of multiply-nested encapsulation, with each layer being peeled off at the end of a tunnel.

The packets in a tunnel-based approach are routed and forwarded in the packet network as normal packets and so this approach is not Semantic Routing.

6.5. Inter-Domain Routing

A lot of effort has been devoted to consideration of end-to-end paths for IP traffic across multiple autonomous systems (ASes). For example, the BGP Add-Paths feature [RFC7911] allows the advertisement of multiple paths so that a single, "best" path can be determined. These approaches, however, are principally concerned with overall reachability, and then with selecting the path with the fewest transit autonomous systems. They are less capable of selecting an overall least cost path or of considering other traffic engineering constraints in the selection of end-to-end paths. Such path

computation requires the features outlined in Section 6.7 as assembled into an architectural solution in [RFC7926].

Many approaches have been suggested [RFC6115] for improving inter-domain routing performance and scaling using address partitioning schemes including tunneling across domains (see also Section 6.4). However, routing in this inter-domain scenario is about the selection of the next AS along the path, and possibly a choice of the right AS border router (ASBR) to facilitate that route. This choice of ASBRs might be based on additional information carried in the packets so could qualify as Semantic Routing, but packets flowing between these ASBRs are routed and forwarded within the domains as normal packets without the use of Semantic Routing.

6.6. Service Function Chaining

Service Function Chaining (SFC) [RFC7665] is applied at the network layer to steer packet flows through network functions (such as security or load balancing). A chain of services to be delivered (the service function chain) is realized as sequence of service instances (the service function path). Packets are tunneled between the service instances using encapsulation so that the end-to-end payload packet is unchanged. A variety of network layer encapsulation have been considered including the Network Service Header (NSH) [RFC8300], MPLS [RFC8595], and Segment Routing [I-D.li-spring-sr-sfc-control-plane-framework].

The Segment Routing concept of Network Programming [RFC8986], offers a similar approach to SFC, but may be more widely applicable.

The tunneled packets can be freely routed in the network using conventional shortest path techniques or the mechanisms described in Section 6.7 and Section 6.8, thus this approach is not Semantic Routing.

6.7. Network Layer Traffic Engineering Techniques

Techniques for achieving packet-level traffic engineering in the network layer are described in [I-D.ietf-teas-rfc3272bis]. Traffic engineering (TE) is the process of selecting an end-to-end path that considers many attributes of metrics of the links in the network in order to satisfy a set of constraints or requirements imposed by the sender of the traffic. For example, the sender may want to use only secure links, or may know the bandwidth requirements of the flow, or may need at least a specific end-to-end latency, or indeed any combination of this type of constraint.

Routing for TE may be performed in advance of sending the traffic (for example, by computing a path at the sender or by using a tool such as the Path Computation Element (PCE) [RFC4655]. In this case, some form of encapsulation is needed to bind the traffic flow to the selected route: MPLS or Segment Routing may be used.

Alternatively, the network may be tuned through appropriate use of routing protocol metrics, routing algorithms, and statically configured routes, so that packets will be forwarded along traffic engineered paths.

6.8. Semantic Routing in the Network Layer

Semantic routing, as already explained, is about taking routing decisions based on "additional" information carried in packets in order to provide the behavior and network services most suited to the traffic. This approach builds on the techniques described in Section 6.7 but frees up the network to make individual decisions for each packet based on changing network conditions as well as the information in the packets.

A raft of potential solutions have been proposed for carrying the necessary information in the packets, and it is not the purpose of this document to examine them in detail or make suggestions about which is better. The solutions vary from simply using existing fields in the IP header (such as the ToS field), or examining fields below the IP header (such as the transport ports), through "overloading" existing fields in the packet header (such as the destination address), all the way to adding new information in an additional encapsulation as proposed by the Application-aware Networking (APN) effort [I-D.li-apn-framework].

6.9. Computation In The Network and Semantic Routing

The use of semantic enhancements as a key aspect to Semantic Routing (as described in this document) links the development of Semantic Routing solutions to data plane programmability. Novel approaches to semantic routing may inform the evolution of more complex in-network operations, aiding specific Semantic Routing solutions. Further, progress in routing protocols (e.g., on multi-optimality routing [SOBRINHO]) may be seen as a key input into the more general problem within an emerging framework to distribute state needed for in-network computing operations, e.g., through utilizing insights from routing protocols to distribute routing state for more limited routing operations.

As per its charter, the Computation In The Network (COIN) Research Group [COINRG] combines the idea of computing with the programmability of the data plane. Hence, network operations, such as those previously used for routing and forwarding, may be key to the programmability aspects of "computing in the network" within the scope of COIN. Ultimately, as stated in the COIN charter, "The goal is to investigate how to harness and to benefit from this emerging disruption to the Internet architecture to improve network and application performance as well as user experience." From this, we can conclude that data plane programmability and its impact on existing and emerging areas of communication are key to COIN.

The COIN charter further states, "COIN specifically will focus on the evolution necessary for networking to move beyond packet interception as the basis of network operation and into computation." This envisions that data plane programmability is not limited to packet interception, but may evolve towards more complex operations on data flowing across the network. The analysis of use cases and the identification of key areas of study can drive the understanding of what those additional operations may be and how to program them, particularly across several participating network elements and at the endpoints. With this, we can conclude that the areas for applying COIN ideas will ultimately drive the evolution of COIN technologies by identifying emerging requirements and uses for data plane programmability, particularly those beyond simple packet processing, such as packet forwarding and local buffer management.

Given the focus on steering traffic between micro-services instantiated at computational elements within networks and at endpoints, the COIN use cases identify aspects of what is now amed Semantic Routing. Thus Semantic Routing is one possible applicability area for COIN.

Conversely, the availability of emerging data plane programmability may enable new capabilities for Semantic Routing. As a distributed problem, Semantic Routing could be enabled by emerging programming frameworks that may be developed within the work of COIN, possibly leading to new ways of orchestrating and deploying distributed routing programs. Thus, the relationship between Semantic Routing and the COIN Research Group can be characterized as a symbiotic process of informing and enabling that may benefit both work areas.

7. Security Considerations

Semantic routing must give full consideration to the security and privacy issues that are introduced by these mechanisms. Placing additional information into packet header fields might reveal details of what the packet is for, what function the user is performing, who the user is, etc. Furthermore, in-flight modification of the additional information might not directly change the destination of the packet, but might change how the packet is handled within the network and at the destination.

It should also be considered how packet encryption techniques that are increasingly popular for end-to-end or edge-to-edge security may obscure the semantic information carried in some fields of the packet header or found deeper in the packet. This may render some semantic routing techniques impractical and may dictate other methods of carrying the necessary information to enable semantic routing.

8. IANA Considerations

This document makes no requests for IANA action.

9. Acknowledgements

Thanks to Brian Carpenter, Dave Oran, and Luigi Iannone for helpful discussions and clarifications.

10. Contributors

Mohamed Boucadair
Email: mohamed.boucadair@orange.com

Dirk Trossen
Email: dirk.trossen@huawei.com

11. References

11.1. Informative References

[BLIND-FORWARDINGref]
Simsek, I., "On-Demand Blind Packet Forwarding", Paper 30th International Telecommunication Networks and Applications Conference (ITNAC), 2020, 2020, <https://www.computer.org/csdl/proceedings-article/itnac/2020/09315187/1qmfFPPggrC>.

[CONTENT-RTG-MOBILEref]
Liu, H. and W. He, "Rich Semantic Content-oriented Routing for mobile Ad Hoc Networks", Paper The International Conference on Information Networking (ICOIN2014), 2014, 2014, <https://ieeexplore.ieee.org/document/6799682>.

[EIBPref] Shenoy, N., "Can We Improve Internet Performance? An Expedited Internet Bypass Protocol", Presentation 28th IEEE International Conference on Network Protocols, 2020, <https://icnp20.cs.ucr.edu/Slides/NIPAA/D-3_invited.pptx>.

[GEO-IPref]
Dasu, T., Kanza, Y., and D. Srivastava, "Geotagging IP Packets for Location-Aware Software-Defined Networking in the Presence of Virtual Network Functions", Paper 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM SIGSPATIAL 2017), 2017, <https://about.att.com/ecms/dam/sites/labs_research/content/publications/AI_Geotagging_IP_Packets_for_Location.pdf>.

[I-D.ietf-teas-rfc3272bis]
Farrel, A., "Overview and Principles of Internet Traffic Engineering", Work in Progress, Internet-Draft, draft-ietf-teas-rfc3272bis-16, 24 March 2022, <https://www.ietf.org/archive/id/draft-ietf-teas-rfc3272bis-16.txt>.

[I-D.jiang-semantic-prefix]
Jiang, S., Sun, Q., Farrer, I., Bo, Y., and T. Yang, "Analysis of Semantic Embedded IPv6 Address Schemas", Work in Progress, Internet-Draft, draft-jiang-semantic-prefix-06, 15 July 2013, <https://www.ietf.org/archive/id/draft-jiang-semantic-prefix-06.txt>.

[I-D.jiang-service-oriented-ip]
Carpenter, B., Jiang, S., and G. Li, "Service Oriented Internet Protocol", Work in Progress, Internet-Draft, draft-jiang-service-oriented-ip-03, 14 May 2020, <https://www.ietf.org/archive/id/draft-jiang-service-oriented-ip-03.txt>.

[I-D.king-irtf-challenges-in-routing]
King, D., Farrel, A., and C. Jacquenet, "Challenges for the Internet Routing Infrastructure Introduced by Semantic Routing", Work in Progress, Internet-Draft, draft-king-irtf-challenges-in-routing-07, 22 January 2022, <https://www.ietf.org/archive/id/draft-king-irtf-challenges-in-routing-07.txt>.

[I-D.king-irtf-semantic-routing-survey]
King, D. and A. Farrel, "A Survey of Semantic Internet Routing Techniques", Work in Progress, Internet-Draft, draft-king-irtf-semantic-routing-survey-03, 26 November 2021, <https://www.ietf.org/archive/id/draft-king-irtf-semantic-routing-survey-03.txt>.

[I-D.li-apn-framework]
Li, Z., Peng, S., Voyer, D., Li, C., Liu, P., Cao, C., and G. Mishra, "Application-aware Networking (APN) Framework", Work in Progress, Internet-Draft, draft-li-apn-framework-05, 7 March 2022, <https://www.ietf.org/archive/id/draft-li-apn-framework-05.txt>.

[I-D.li-spring-sr-sfc-control-plane-framework]
Li, C., Sawaf, A. E., Hu, R., and Z. Li, "A Framework for Constructing Service Function Chaining Systems Based on Segment Routing", Work in Progress, Internet-Draft, draft-li-spring-sr-sfc-control-plane-framework-06, 21 April 2022, <https://www.ietf.org/archive/id/draft-li-spring-sr-sfc-control-plane-framework-06.txt>.

[IEN23] Cohen, D., "IEN 23: On Names, Addresses and Routings", Internet Experiment Note IEN 23, Notebook Section 2.3.3.7, 1978, <https://www.rfc-editor.org/ien/ien23.pdf>.

[MULTICAST-SRref]
Jia, W. and W. He, "A Scalable Multicast Source Routing Architecture for Data Center Networks", Paper IEEE Journal on Selected Areas in Communications, vol. 32, no. 1, pp. 116-123, January 2014, 2014, <https://ieeexplore.ieee.org/document/6799682>.

[OPENSRNref]
Ren, P., Wang, X., Zhao, B., Wu, C., and H. Sun, "OpenSRN: A Software-defined Semantic Routing Network Architecture", Paper IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Hong Kong, 2015, 2015, <https://www.researchgate.net/publication/308827498_OpenSRN_A_software-defined_semantic_routing_network_architecture>.

[RFC0814] Clark, D., "Name, addresses, ports, and routes", RFC 814, DOI 10.17487/RFC0814, July 1982, <https://www.rfc-editor.org/info/rfc814>.

[RFC1853] Simpson, W., "IP in IP Tunneling", RFC 1853, DOI 10.17487/RFC1853, October 1995, <https://www.rfc-editor.org/info/rfc1853>.

[RFC3135] Border, J., Kojo, M., Griner, J., Montenegro, G., and Z. Shelby, "Performance Enhancing Proxies Intended to Mitigate Link-Related Degradations", RFC 3135, DOI 10.17487/RFC3135, June 2001, <https://www.rfc-editor.org/info/rfc3135>.

[RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, DOI 10.17487/RFC4291, February 2006, <https://www.rfc-editor.org/info/rfc4291>.

[RFC4655] Farrel, A., Vasseur, J.-P., and J. Ash, "A Path Computation Element (PCE)-Based Architecture", RFC 4655, DOI 10.17487/RFC4655, August 2006, <https://www.rfc-editor.org/info/rfc4655>.

[RFC6052] Bao, C., Huitema, C., Bagnulo, M., Boucadair, M., and X. Li, "IPv6 Addressing of IPv4/IPv6 Translators", RFC 6052, DOI 10.17487/RFC6052, October 2010, <https://www.rfc-editor.org/info/rfc6052>.

[RFC6115] Li, T., Ed., "Recommendation for a Routing Architecture", RFC 6115, DOI 10.17487/RFC6115, February 2011, <https://www.rfc-editor.org/info/rfc6115>.

[RFC6282] Hui, J., Ed. and P. Thubert, "Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks", RFC 6282, DOI 10.17487/RFC6282, September 2011, <https://www.rfc-editor.org/info/rfc6282>.

[RFC6992] Cheng, D., Boucadair, M., and A. Retana, "Routing for IPv4-Embedded IPv6 Packets", RFC 6992, DOI 10.17487/RFC6992, July 2013, <https://www.rfc-editor.org/info/rfc6992>.

[RFC7285] Alimi, R., Ed., Penno, R., Ed., Yang, Y., Ed., Kiesel, S., Previdi, S., Roome, W., Shalunov, S., and R. Woundy, "Application-Layer Traffic Optimization (ALTO) Protocol", RFC 7285, DOI 10.17487/RFC7285, September 2014, <https://www.rfc-editor.org/info/rfc7285>.

[RFC7597] Troan, O., Ed., Dec, W., Li, X., Bao, C., Matsushima, S., Murakami, T., and T. Taylor, Ed., "Mapping of Address and Port with Encapsulation (MAP-E)", RFC 7597, DOI 10.17487/RFC7597, July 2015, <https://www.rfc-editor.org/info/rfc7597>.

[RFC7665] Halpern, J., Ed. and C. Pignataro, Ed., "Service Function Chaining (SFC) Architecture", RFC 7665, DOI 10.17487/RFC7665, October 2015, <https://www.rfc-editor.org/info/rfc7665>.

[RFC7761] Fenner, B., Handley, M., Holbrook, H., Kouvelas, I., Parekh, R., Zhang, Z., and L. Zheng, "Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)", STD 83, RFC 7761, DOI 10.17487/RFC7761, March 2016, <https://www.rfc-editor.org/info/rfc7761>.

[RFC7911] Walton, D., Retana, A., Chen, E., and J. Scudder, "Advertisement of Multiple Paths in BGP", RFC 7911, DOI 10.17487/RFC7911, July 2016, <https://www.rfc-editor.org/info/rfc7911>.

[RFC7926] Farrel, A., Ed., Drake, J., Bitar, N., Swallow, G., Ceccarelli, D., and X. Zhang, "Problem Statement and Architecture for Information Exchange between Interconnected Traffic-Engineered Networks", BCP 206, RFC 7926, DOI 10.17487/RFC7926, July 2016, <https://www.rfc-editor.org/info/rfc7926>.

[RFC8300] Quinn, P., Ed., Elzur, U., Ed., and C. Pignataro, Ed., "Network Service Header (NSH)", RFC 8300, DOI 10.17487/RFC8300, January 2018, <https://www.rfc-editor.org/info/rfc8300>.

[RFC8595] Farrel, A., Bryant, S., and J. Drake, "An MPLS-Based Forwarding Plane for Service Function Chaining", RFC 8595, DOI 10.17487/RFC8595, June 2019, <https://www.rfc-editor.org/info/rfc8595>.

[RFC8799] Carpenter, B. and B. Liu, "Limited Domains and Internet Protocols", RFC 8799, DOI 10.17487/RFC8799, July 2020, <https://www.rfc-editor.org/info/rfc8799>.

[RFC8986] Filsfils, C., Ed., Camarillo, P., Ed., Leddy, J., Voyer, D., Matsushima, S., and Z. Li, "Segment Routing over IPv6 (SRv6) Network Programming", RFC 8986, DOI 10.17487/RFC8986, February 2021, <https://www.rfc-editor.org/info/rfc8986>.

[RFC9139] Gündoan, C., Schmidt, T., Wählisch, M., Scherb, C., Marxer, C., and C. Tschudin, "Information-Centric Networking (ICN) Adaptation to Low-Power Wireless Personal Area Networks (LoWPANs)", RFC 9139, DOI 10.17487/RFC9139, November 2021, <https://www.rfc-editor.org/info/rfc9139>.

[SEMANTICRTG] Strassner, J., Sung-Su, K., and J. Won-Ki, "Semantic Routing for Improved Network Management in the Future Internet", Book Chapter Springer, Recent Trends in Wireless and Mobile Networks, 2010, 2010, <https://link.springer.com/chapter/10.1007/978-3-642-14171-3_14>.

[SOBRINHO] Sobrinho, J. and M. Ferreira, "Routing on Multiple Optimality Criteria", Paper SIGCOMM 2020, 2020, <https://dl.acm.org/doi/abs/10.1145/3387514.3405864>.

[TERASTREAMref] Zaluski, B., Rajtar, B., Habjani, H., Baranek, M., Slibar, N., Petracic, R., and T. Sukser, "Terastream implementation of all IP new architecture", Paper 36th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2013, 2013, <https://ieeexplore.ieee.org/document/6596297>.

11.2. URL References

[COINRG] Internet Engineering Research Group, "Computation in the Network Research Group (COINRG)", 2022, <https://irtf.org/coinrg>.

[ONION] The Tor Project, Inc., "The Onion Routing Project : Anonymity Online", 2022, <https://torproject.org/>.

Authors' Addresses

Adrian Farrel
Old Dog Consulting
United Kingdom
Email: adrian@olddog.co.uk

Daniel King
Lancaster University
United Kingdom
Email: d.king@lancaster.ac.uk

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: 3 August 2024

A. Choudhary
Aviatrix Systems
M. Jethanandani
Kloud Services
E. Aries
Juniper Networks
I. Chen
The MITRE Corporation
31 January 2024

YANG Models for Quality of Service (QoS)
draft-ietf-rtgwg-qos-model-12

Abstract

This document describes a YANG model for configuration and operational data of Quality of Service (QoS) in network devices.

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 3 August 2024.

Table of Contents

1. Introduction

This document defines a YANG 1.1 [RFC7950] model for Quality of Service (QoS) configuration and statistics parameters. Traffic Policy module defines the basic building blocks to define a classifier, policy and target. QoS Action module defines QoS action related parameters. Traffic policy is augmented to include packet match and action parameters to define the Differentiated Services (DiffServ) policy, Queues policy and Scheduler policy. Each of these policies are defined as separate modules. The DiffServ module is based on DiffServ architecture, and various references have been made to available standard architecture documents. QoS statistics counters are defined in QoS Operational module.

DiffServ is a preferred approach for network service providers to offer services to different customers based on their network Quality-of-Service (QoS) objectives. The traffic streams are differentiated based on DiffServ Code Points (DSCP) carried in the IP header of each packet. The DSCP markings are applied by upstream node or by the edge router on entry to the DiffServ network.

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

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

1.1. Note to RFC Editor

Editorial Note: (To be removed by RFC Editor)

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

* "XXXX" --> the assigned RFC value for this draft both in this draft and in the yang modules under the revision statement.

* The "revision" date in model, in the format XXXX-XX-XX, needs to be updated with the date the draft gets approved.

1.2. Terminology

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

1.3. Definitions and Acronyms

This document uses definitions and acronyms defined in Definitions of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers [RFC2474], An Architecture for Differentiated Services [RFC2475], and other documents. Here are some of them.

* Active Queue Management (AQM): An activity that marks or drops packets before the queue is full.

* Behavior Aggregate (BA): A DS behavior aggregate.

* Classifier: an entity which selects packets based on the content of packet headers according to defined rules.

* DS behavior aggregate: A collection of packets with the same DS codepoint crossing a link in a particular direction.

* DS code point: A specific value of the DSCP portion of the DS field, used to select a PHB.

* DiffServ: Differentiated Services enhancements to the Internet protocol are intended to enable scalable service discrimination in the Internet without the need for per-flow state and signaling at every hop. A variety of services may be built from a small, well-defined set of building blocks which are deployed in network nodes.

* DSCP: Differentiated Services Code Point

* ECN: Explicit Congestion Notification

* Marking: the process of setting the DS codepoint in a packet based on defined rules; pre-marking, re-marking.

* Metering: the process of measuring the temporal properties (e.g., rate) of a traffic stream selected by a classifier. The instantaneous state of this process may be used to affect the operation of a marker, shaper, or dropper, and/or may be used for accounting and measurement purposes.

* MF Classifier: A multi-field (MF) classifier which selects packets based on the content of some arbitrary number of header fields; typically some combination of source address, destination address, DS field, protocol ID, source port and destination port.

* Per-Hop-Behavior (PHB): The externally observable forwarding behavior applied at a DS-compliant node to a DS behavior aggregate.

* Policing: the process of discarding packets (by a dropper) within a traffic stream in accordance with the state of a corresponding meter enforcing a traffic profile.

* RED: Random Early Detection

* Shaping: the process of delaying packets within a traffic stream to cause it to conform to some defined traffic profile.

* Traffic-Group: is a metadata stored in a packet buffer.

* WRED: Weighted Random Early Detection

2. QoS Model Design

The overall objective of this model design is to offer a base set of modules that can then be further augmented to provide other models, such as Differentiated Services (DiffServ) and Policy Based Routing (PBR). Those models would augment the base modules defined in this draft to define their own model. This document defines one such model, DiffServ. The design also takes into consideration that vendors might want to extend the model to add their own extensions, such as to extend Traffic Policy module, to provide constructs of policy for different QoS functionalities."

```
 -------------   ------------   ------------   ------------
|             | |            | |            | |            |
| DiffServ    | | Queue      | | Scheduling | | QoS        |
| Policy      | | Policy     | | Policy     | | Oper       |
|             | |            | |            | |            |
 -------------   ------------   ------------   ------------
 -------------   ------------------------------------------
|             | |                                          |
| QoS Action  | |  Traffic Policy                          |
|             | | (Base Policy, Classifier, Target)        |
|             | |                                          |
 -------------   ------------------------------------------
```

Figure 1: QoS Yang Modules Layout

The above diagram depicts the design of the QoS model. It uses Traffic Policy and QoS Action as base modules, which are augmented by the remaining modules on the top to define different aspects of QoS.

The Classifier is defined as part of Traffic Policy Module. A classifier consists of set of filters and an operation that is performed on those filters. Each filter defines a rule on a particular packet header or packet metadata field that dictate how the packet will be classified based on that particular field, e.g. whether it will be classified based on DSCP, or will be classified based on source IPv4 address prefix. The operation defines whether these filters are logically AND or OR. The filters may be based on a combination of values or range of values of different packet header fields or packet metadata fields. An example of how these filter definitions are used can be seen in the DiffServ model. The DiffServ model augments the filter node in the Classifier to add in DiffServ specific filter parameters.

One or more packet conditioning functions may be applied to a classifier which may drop, mark, or delay packets. These are defined in the QoS Action module. A set of classifiers with corresponding conditioning functions when arranged in order of priority represent a QoS policy. These policies are defined in Traffic Policy module.

The QoS Action module defines marking, metering and queuing. QoS actions are configured in line or referred to in DiffServ, Queuing, and Scheduling Policy modules.

A Meter meters each packet and passes the packet and the metering result to the Marker. Meter is modeled based on commonly used algorithms in industry, A Single Rate Three Color Marker (srTCM) [RFC2697], A Two Rate Three Color Marker (trTCM) [RFC2698], and A Single Rate Two Color Marker. Different vendors can extend it with other types of meters as well. Meter is defined in QoS Action module.

Queue Policy module allows queues to be referred in a policy. The match is based on traffic group and action parameters are used as defined in QoS Action module.

Scheduling Policy module defines a set of scheduling parameters and associates Queue Policy with it.

QoS counters are defined in QoS Operational module. It includes counters for Classifiers, Meters and Queues associated with a QoS policy applied in each direction of traffic. To modularize and for reusability, grouping have been defined for various counters of Classifier, Meters and Queues. The target is assumed to be interface but the groupings can be used for any other target type where QoS policy is applied.

3. DiffServ Model Design

Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers [RFC2475] describes the architecture as a simple model where traffic entering a network is classified and possibly conditioned at the boundary of the network and assigned a different Behavior Aggregate (BA). Each BA is identified by a specific value of DSCP, and is used to select a Per Hop Behavior (PHB).

The packet classification policy identifies the subset of traffic which may receive a DiffServ by being conditioned or mapped. Packet classifiers select packets within a stream based on the content of some portion of the packet header. There are two types of classifiers, the BA classifier, and the Multi-Field (MF) classifier which selects packets based on a value which is combination of one or more header fields. In the DiffServ Policy module, this is realized by augmenting the classifier in Traffic Policy module.

Traffic conditioning includes metering, shaping and/or marking. A Meter is used to measure the traffic against a given traffic profile. The traffic profile specifies the temporal property of the traffic. A packet that arrives is first determined to be in or out of the profile, which will result in the action of marked, dropped or shaped. This is realized in vendor specific modules based on the parameters defined in action module. The metering parameters are augmented to the QoS policy module when metering is defined inline, and to the metering template when metering profile is referred in policy module.

3.1. A Single Rate Three Color Marker

This document defines support for A Single Rate Three Color Marker (srTCM) [RFC2697], which is one of the components in a An Architecture for Differentiated Services [RFC2475]. The srTCM meters a traffic stream and marks its packets according to three traffic parameters, Committed Information Rate (CIR), Committed Burst Size (CBS), and Excess Burst Size (EBS), to be either green, yellow, or red. A packet is marked green if it doesn’t exceed the CBS, yellow if it does exceed the CBS, but not the EBS, and red otherwise.

3.2. A Two Rate Three Color Marker

This document defines support for A Two Rate Three Color Marker (trTCM) [RFC2698], which is one of the components in a An Architecture for Differentiated Services [RFC2475]. The trTCM meters a traffic stream and marks its packets according to two rates, Peak Information Rate (PIR), the Committed Information Rate (CIR), and their associated burst sizes to either green, yellow, or red. A packet is marked red if it exceeds the PIR. Otherwise, it is marked yellow or green depending on whether it exceeds or does not exceed the CIR.

4. YANG Modules

Modules defined in this draft import definitions from Common YANG Data Types [RFC6991] and A YANG Data Model for Interface Management [RFC8343].

4.1. Traffic Policy Module

Traffic Policy module contains list of classifiers identified by a classifier name. Each classifier MAY contain a list of filters.

This module also contains list of policy objects identified by a policy name and policy type which MUST be provided. Policy-type is of type identity and is populated in a vendor specific manner. With different values of policy types, each vendor MAY define their own construct of policy for different QoS functionalities. Each vendor MAY augment classifier entry in a policy definition with a set of actions.

The module also augments A YANG Data Model for Interface Management [RFC8343] module to add a target policy. A single policy of a particular policy-type can be applied on an interface in each direction of traffic.

```
module: ietf-traffic-policy
  +--rw classifiers
  |  +--rw classifier* [name]
  |     +--rw name                string
  |     +--rw description?        string
  |     +--rw filter-operation?   qos-types:match-operation-type
  |     +--rw filter* [type logical-not]
  |           ...
  +--rw policies
     +--rw policy* [name type]
        +--rw name           string
        +--rw type           identityref
        +--rw description?   string
        +--rw classifier* [name]
              ...

  augment /if:interfaces/if:interface:
    +--rw qos-target-policy* [direction type]
       +--rw direction    identityref
       +--rw type         identityref
       +--rw name         string
```

Figure 2: ietf-traffic-policy abridged tree diagram

```
<CODE BEGINS> file "ietf-traffic-policy@2024-01-31.yang"
module ietf-traffic-policy {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-traffic-policy";
  prefix traffic-policy;

  import ietf-interfaces {
    prefix if;
  }
  import iana-qos-types {
    prefix qos-types;
    reference
      "RFC XXXX: YANG Data Models for Quality of Service (QoS).";
  }

  organization
    "IETF Routing Area Working Group.";

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

     Editor:   Aseem Choudhary
               <mailto:achoudhary@aviatrix.com>
```

```
        Editor:   Mahesh Jethanandani
                  <mailto:mjethanandani@gmail.com>";

     description
       "This module contains a collection of YANG definitions for
        configuring QoS Traffic Policies.

        Copyright (c) 2023 IETF Trust and the persons identified as
        authors of the code.  All rights reserved.

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

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

     revision 2024-01-31 {
       description
         "Initial version.";
       reference
         "RFC XXXX: YANG Models for Quality of Service (QoS).";
     }

     grouping filters {
       description
         "Filters types in a Classifier entry.";

       leaf type {
         type identityref {
           base qos-types:filter-type;
         }
         description
           "This leaf defines the type of filter.";
       }

       leaf logical-not {
         type boolean;
         description
           "This is logical-not operator for filter values and/or value
            ranges. When true, it indicates filter looks for absence
            of values and/or value ranges defined by the filter.";
       }
     }
```

```
     grouping generic-classifier-attr {
       description
         "Generic Classifier attributes like name,
          description, operation type.";

       leaf name {
         type string;
         description
           "A unique name for Classifier entry.";
       }

       leaf description {
         type string;
         description
           "Description statement for classifier entry.";
       }

       leaf filter-operation {
         type qos-types:match-operation-type;
         description
           "match-any or match-all filters.";
       }
     }

     grouping inline-attr {
       description
         "Attributes of inline classifier in a policy.";

       leaf filter-operation {
         type qos-types:match-operation-type;
         description
           "match-any or match-all filters.";
       }

       list filter {
         key "type logical-not";
         uses filters;
         description
           "Filters configured inline in a policy.";
       }
     }

     container classifiers {
       description
         "List of classifier entries.";

       list classifier {
         key "name";
```

```
        description
          "Each classifier entry contains a list of filters.";

        uses generic-classifier-attr;

        list filter {
          key "type logical-not";
          uses filters;
          description
            "Filter entries for a classifier.";
        }
      }
    }

    container policies {
      description
        "List of policy templates.";

      list policy {
        key "name type";
        description
          "Policy template.";

        leaf name {
          type string;
          description
            "A unique name for the policy.";
        }

        leaf type {
          type identityref {
            base qos-types:policy-type;
          }
          description
            "The type of policy.";
        }

        leaf description {
          type string;
          description
            "A description for policy.";
        }

        list classifier {
          key "name";
          ordered-by user;
          description
            "Classifier configuration in a policy.";
```

```
          leaf name {
            type string;
            description
              "A unique name for classifier entry.";
          }

          container inline {
            presence "This container is present for when the device
                      is configured for inline classification.";
            uses inline-attr;
            description
              "Container for when an inline classification
               is desired.";
          }

          list action {
            key "type";
            ordered-by user;
            description
              "List of action for a Classifier.";

            leaf type {
              type identityref {
                base qos-types:action-type;
              }
              description
                "Type of action.";
            }

          }
        }
      }
    }

    augment "/if:interfaces/if:interface" {
      description
        "Augments Interface module to add Target Entry.";

      list qos-target-policy {
        key "direction type";
        description
          "Policy target for inbound or outbound direction.";

        leaf direction {
          type identityref {
            base qos-types:direction;
          }
          description
```

```
            "Direction of the traffic flow either inbound or
             outbound.";
        }

        leaf type {
          type identityref {
            base qos-types:policy-type;
          }
          description
            "Policy type.";
        }

        leaf name {
          type string;
          mandatory true;
          description
            "A unique name for the Policy.";
        }
      }
    }
  }
  <CODE ENDS>
```

Figure 3: ietf-traffic-policy module

4.2. QoS Action Module

QoS Action module includes actions such marking, metering, or queuing. Metering, marking, queuing, and scheduling actions are realized by augmenting the Traffic Policy module. Marking sets DiffServ codepoint value in the classified packet.

```
module: ietf-qos-action
  +--rw meters
  |  +--rw meter* [name]
  |     +--rw name                                    string
  |     +--rw (meter-types)?
  |           ...
  +--rw queues
     +--rw queue* [name]
        +--rw name     string
        +--rw queue
              ...
```

Figure 4: ietf-qos-actions abridged tree diagram

```
<CODE BEGINS>
file "ietf-qos-action@2024-01-31.yang"
module ietf-qos-action {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-qos-action";
  prefix qos-action;

  import ietf-inet-types {
    prefix inet;
    reference
      "RFC 6991: Common YANG Data Types";
  }
  import iana-qos-types {
    prefix qos-types;
    reference
      "RFC XXXX: YANG Model for QoS";
  }

  organization
    "IETF Routing Area Working Group";

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

     Editor:   Aseem Choudhary
               <mailto:achoudhary@aviatrix.com>
     Editor:   Mahesh Jethanandani
               <mailto:mjethanandani@gmail.com>";

  description
    "This module contains a collection of YANG definitions for
     configuring QoS actions.

     Copyright (c) 2023 IETF Trust and the persons identified as
     authors of the code.  All rights reserved.

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

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

  revision 2024-01-31 {
```

```
      description
        "Initial version";
      reference
        "RFC XXXX: YANG Models for Quality of Service (QoS).";
    }

    grouping rate-value-unit {
      leaf value {
        type uint64;
        description
          "Value for the rate that is configured, either in
           bits-per-second or percentage.";
      }

      leaf unit {
        type identityref {
          base qos-types:rate-unit-type;
        }
        default qos-types:bits-per-second;
        description
          "The unit for the value above, either in bits-per-second or
           percentage.";
      }
      description
        "Grouping for configuration of value and unit.";
    }

    grouping meter-action {
      description
        "Configuration of meter actions.";

      leaf action-type {
        type identityref {
          base qos-types:meter-action-type;
        }
        description
          "Action type for metering.";
      }

      leaf dscp-mark {
        type inet:dscp;
        must "../action-type = 'qos-types:action-mark'" {
          error-message
            "DSCP must be marked when action-type is action-mark.";
        }
        description
          "DSCP marking";
      }
```

```
      leaf traffic-group-mark {
        type string;
        must "../action-type = 'qos-types:action-mark'" {
          error-message
            "Traffic-group must be marked when action-type is
             action-mark.";
        }
        description
          "A unique name to mark as a traffic group.";
      }
    }

    grouping single-rate-two-color-meter {
      container single-rate-two-color-meter {
        description
          "A container for Basic Single Rate Two Color Marker.";

        leaf committed-information-rate {
          type uint64;
          units "bytes-per-second";
          description
            "The value for Committed Information Rate (CIR).";
        }

        leaf committed-burst-size {
          type uint64;
          units "bytes";
          description
            "The value for Committed Burst Size (CBS).";
        }

        container conform-action {
          uses meter-action;
          description
            "A packet is marked conforming, or green if it does not
             exceed CIR.";
        }

        container exceed-action {
          uses meter-action;
          description
            "A packet is marked exceeding, or yellow if it does
             exceed CIR.";
        }
      }
      description
        "A Basic Single Rate Two Color Meter.";
    }
```

```
     grouping single-rate-three-color-meter {
       container single-rate-three-color-meter {
         description
           "Container that defines the Single Rate Three Color Meter
            (srTCM).";

         leaf committed-information-rate {
           type uint64;
           units "bytes-per-second";
           description
             "The value for Committed Information Rate (CIR).";
           reference
             "RFC 2697: A Single Rate Three Color Marker.";
         }

         leaf committed-burst-size {
           type uint64;
           units "bytes";
           description
             "The value for Committed Burst Size (CBS).";
           reference
             "RFC 2697: A Single Rate Three Color Marker.";
         }

         leaf excess-burst-size {
           type uint64;
           units "bytes";
           description
             "The value for Excess Burst Size (EBS).";
           reference
             "RFC 2697: A Single Rate Three Color Marker.";
         }

         container conform-action {
           uses meter-action;
           description
             "A packet is marked conforming, or green if it does not
              exceed CBS.";
           reference
             "RFC 2697: A Single Rate Three Color Marker.";
         }

         container exceed-action {
           uses meter-action;
           description
             "A packet is marked exceeding, or yellow if it does
              exceed CBS, but not the EBS.";
           reference
```

```
            "RFC 2697: A Single Rate Three Color Marker.";
        }

        container violate-action {
          uses meter-action;
          description
            "A packet is marked violating, or red if it is exceeds
             both CBS and EBS.";
          reference
            "RFC 2697: A Single Rate Three Color Marker.";
        }
      }
      description
        "Singe Rate Three Color Meter (srTCM).";
      reference
        "RFC 2697: A Single Rate Three Color Marker.";
    }

    grouping two-rate-three-color-meter {
      container two-rate-three-color-meter {
        description
          "A container that defines A Two Rate Three Color Meter
           (trTCM).";

        leaf committed-information-rate {
          type uint64;
          units "bytes-per-second";
          description
            "The value for Committed Information Rate (CIR).";
          reference
            "RFC 2698: A Two Rate Three Color Marker.";
        }

        leaf committed-burst-size {
          type uint64;
          units "bytes";
          description
            "The value for Committed Burst Size (CBS).";
          reference
            "RFC 2698: A Two Rate Three Color Marker.";
        }

        leaf peak-information-rate {
          type uint64;
          units "bytes-per-second";
          description
            "The value for Peak Information Rate (PIR).";
          reference
```

```
            "RFC 2698: A Two Rate Three Color Marker.";
        }

        leaf peak-burst-size {
          type uint64;
          units "bytes";
          description
            "The value for Peak Burst Size (PBS).";
          reference
            "RFC 2698: A Two Rate Three Color Marker.";
        }

        container conform-action {
          uses meter-action;
          description
            "A packet is marked conforming, or green, if it does not
             exceed CIR.";
          reference
            "RFC 2698: A Two Rate Three Color Marker.";
        }

        container exceed-action {
          uses meter-action;
          description
            "A packet is marked exceeding, or yellow, if it exceeds
             CIR but not PIR.";
          reference
            "RFC 2698: A Two Rate Three Color Marker.";
        }

        container violate-action {
          uses meter-action;
          description
            "A packet is marked as violating, or red, if it exceeds
             both CIR and PIR.";
          reference
            "RFC 2698: A Two Rate Three Color Marker.";
        }
      }
      description
        "Container for a Two Rate Three Color Marker.";
    }

    grouping meter {
      choice meter-types {
        case single-rate-two-color-meter-type {
          uses single-rate-two-color-meter;
        }
```

```
        case single-rate-three-color-meter-type {
          uses single-rate-three-color-meter;
        }

        case two-rate-three-color-meter-type {
          uses two-rate-three-color-meter;
        }
        description
          "A meter action based on choice of meter action type.";
      }
      description
        "Meter attributes.";
    }

    container meters {
      description
        "List of meter templates.";

      list meter {
        key "name";
        description
          "Meter entry template.";

        leaf name {
          type string;
          description
            "A unique string to identify the meter.";
        }
        uses meter;
      }
    }

    grouping meter-reference {
      container meter {
        leaf name {
          type string ;
          mandatory true;
          description
            "This leaf defines name of the meter referenced.";
        }

        leaf type {
          type identityref {
            base qos-types:meter-type;
          }
          mandatory true;
          description
            "This leaf defines type of the meter.";
```

```
        }
        description
          "The name and type for meter reference.";
      }
      description
        "Grouping to define meter reference.";
    }

    grouping count {
      container count {
      if-feature qos-types:count;

        leaf count-action {
          type empty;
          description
            "Take an action if this is defined.";
        }
        description
          "Container for whether to take an action for count.";
      }
      description
        "The count action grouping.";
    }

    grouping named-counter {
      container named-counter {
      if-feature qos-types:named-counter;
        leaf count-name-action {
          type string;
          description
            "A unique name for named counter.";
        }
        description
          "The named counter action.";
      }
      description
        "The named counter action grouping.";
    }

    grouping discard {
      container discard {
        leaf discard {
          type empty;
          description
            "If defined this enables discard action.";
        }
        description
          "Container for discard action.";
```

```
      }
      description
        "Grouping for discard action.";
    }

    grouping priority {
      container priority {
        leaf level {
          type uint8;
          description
            "Priority level.";
        }
        description
          "Container for priority.";
      }
      description
        "Grouping for priority.";
    }

    grouping min-rate {
      container min-rate {
        uses rate-value-unit;
        description
          "Minimum guaranteed bandwidth.";
      }
      description
        "Grouping for minimum rate.";
    }

    grouping dscp-marking {
      container dscp {
        leaf dscp {
          type inet:dscp;
          description
            "DSCP marking.";
        }
        description
          "Container for DSCP marking.";
      }
      description
        "Grouping for DSCP marking.";
    }

    grouping traffic-group-marking {
      container traffic-group {
        leaf traffic-group {
          type string;
          description
```

```
            "A unique name for traffic group.";
        }
        description
          "Container for traffic group marking container.";
      }
      description
        "Grouping for traffic group marking.";
    }

    grouping child-policy {
      container child-policy {
        if-feature qos-types:child-policy;
        leaf name {
          type string;
          description
            "A unique name for hierarchical policy.";
        }
        description
          "Hierarchical Policy configuration container.";
      }
      description
        "Grouping of Hierarchical Policy configuration.";
    }

    grouping max-rate {
      container max-rate {
        uses rate-value-unit;

        leaf burst-value {
          type uint64;
          description
            "Burst value in bytes.";
        }

        leaf burst-unit {
          type identityref {
            base qos-types:burst-unit-type;
          }
          default qos-types:bytes;
          description
            "Burst value in bytes or milliseconds, with a default
             of bytes.";
        }
        description
          "Maximum rate attributes container.";
      }
      description
        "Grouping for maximum rate attributes.";
```

```
     }

     grouping red-config-parameters {
       leaf min-threshold-val {
         type uint64;
         description
           "Minimum value of RED threshold.";
       }

       leaf min-threshold-unit {
         type identityref {
           base qos-types:red-threshold-unit;
         }
         default qos-types:red-threshold-bytes;
         description
           "The unit for minimum RED threshold value, with the default
            of bytes.";
       }

       leaf max-threshold-val {
         type uint64;
         description
           "Maximum value of RED threshold.";
       }

       leaf max-threshold-unit {
         type identityref {
           base qos-types:red-threshold-unit;
         }
         default qos-types:red-threshold-bytes;
         description
           "The unit for maximum RED threshold value, with the default
            of bytes.";
       }

       leaf weight {
         type uint8;
         description
           "The decay factor for the average queue size
            calculation. The numbers is a 2's exponent.";
       }

       leaf max-probability-val {
         type uint64;
         description
           "Value of maximum probability value. This value need
            be interpreted along with max-probability-unit";
       }
```

```
      leaf max-probability-unit {
        type identityref {
          base qos-types:probability-unit;
        }
        default qos-types:probability-pc;
        description
         "Probability unit type, with a default of percentage.";
      }
      description
        "Random Early Detect (RED) configuration parameters.";
    }

    grouping queue {
      container queue {
        uses priority;
        uses min-rate;
        uses max-rate;
        container algorithmic-drop {
          choice drop-algorithm {
            case tail-drop {
              container tail-drop {
                leaf tail-drop {
                  type empty;
                  description
                    "When defined, tail drop algorithm is enabled.";
                }
                description
                  "Tail Drop configuration container.";
              }
              description
                "Tail Drop choice.";
            }

            case red {
              container red {
                uses red-config-parameters;
                leaf ecn-enabled {
                  type boolean;
                  default "false";
                  description
                    "ECN is enabled on the queue.";
                }
                description
                  "Random Early Detect (RED) configuration.";
              }
            }

            case wred {
```

```
               container wred {
                 list wred {
                   key "profile";

                   leaf profile {
                     type uint8;
                     description
                       "Profile id of each WRED profile.";
                   }

                   leaf color-type {
                     type identityref {
                       base qos-types:wred-color-type;
                     }
                     default qos-types:wred-color-dscp;
                     description
                       "WRED color-type of each profile.";
                   }

                   list color-val {
                     key "min max";

                     leaf min {
                       type uint8;
                       description
                         "Minimum value of color types.";
                     }

                     leaf max {
                       type uint8;
                       description
                         "Maximum value of color types.";
                     }
                     description
                       "List of color markings which constitute
                        a traffic profile.";
                   }
                   uses red-config-parameters;
                   description
                     "List of RED profiles each with its own
                      threshold values.";
                 }

                 leaf ecn-enabled {
                   type boolean;
                   default "false";
                   description
                     "ECN is enabled on the queue.";
```

```
                }
                description
                  "Weighted Random Early Detect (WRED) configuration.";
              }
            }
            description
              "Choice of Drop Algorithm.";
          }
          description
            "Algorithmic Drop configuration container.";
        }
        description
          "Queue configuration container.";
      }
      description
        "Queue grouping.";
    }

    container queues {
      description
        "List of queue templates.";

      list queue {
        key "name";
        description
          "Queue entry template.";

        leaf name {
          type string;
          description
            "A unique name identifying this queue template";
        }
        uses queue;
      }
    }

    grouping queue-reference {
      container queue-reference {
        leaf queue-name {
          type string;
          mandatory true;
          description
            "This leaf defines name of the  queue template
             referenced.";
        }
        description
          "Queue template reference.";
      }
```

```
      description
        "Queue template reference grouping.";
    }

    grouping scheduler {
      container scheduler {
        uses min-rate;
        uses max-rate;
        description
          "Scheduler configuration container.";
      }
      description
        "Scheduler configuration grouping.";
    }
  }
  <CODE ENDS>
```

Figure 5: ietf-qos-actions module

4.3. DiffServ Module

DiffServ module augments Traffic Policy module to enable diffrentiated services for different types of packets.

```
module: ietf-diffserv

  augment /traffic-policy:classifiers/traffic-policy:classifier
            /traffic-policy:filter:
    +--rw (filter-param)?
       +--:(dscp)
       |  +--rw dscp* [min max]
       |        ...
       +--:(source-ipv4-prefix)
       |  +--rw source-ipv4-prefix* [prefix]
       |        ...
       +--:(destination-ipv4-prefix)
       |  +--rw destination-ipv4-prefix* [prefix]
       |        ...
       +--:(source-ipv6-prefix)
       |  +--rw source-ipv6-prefix* [prefix]
       |        ...
       +--:(destination-ipv6-prefix)
       |  +--rw destination-ipv6-prefix* [prefix]
       |        ...
       +--:(source-port)
       |  +--rw source-port* [min max]
       |        ...
       +--:(destination-port)
```

```
          |  +--rw destination-port* [min max]
          |        ...
          +--:(protocol)
          |  +--rw protocol* [min max]
          |        ...
          +--:(traffic-group)
          |  +--rw traffic-group
          |        ...
          +--:(filter-match-all)
             +--rw match-all
                   ...
     augment /traffic-policy:policies/traffic-policy:policy
               /traffic-policy:classifier/traffic-policy:inline
               /traffic-policy:filter:
       +--rw (filter-params)?
          +--:(dscp)
          |  +--rw dscp* [min max]
          |        ...
          +--:(source-ipv4-prefix)
          |  +--rw source-ipv4-prefix* [prefix]
          |        ...
          +--:(destination-ipv4-prefix)
          |  +--rw destination-ipv4-prefix* [prefix]
          |        ...
          +--:(source-ipv6-prefix)
          |  +--rw source-ipv6-prefix* [prefix]
          |        ...
          +--:(destination-ipv6-prefix)
          |  +--rw destination-ipv6-prefix* [prefix]
          |        ...
          +--:(source-port)
          |  +--rw source-port* [min max]
          |        ...
          +--:(destination-port)
          |  +--rw destination-port* [min max]
          |        ...
          +--:(protocol)
          |  +--rw protocol* [min max]
          |        ...
          +--:(traffic-group)
          |  +--rw traffic-group
          |        ...
          +--:(filter-match-all)
             +--rw match-all
                   ...
     augment /traffic-policy:policies/traffic-policy:policy
               /traffic-policy:classifier/traffic-policy:action:
       +--rw (action-params)?
```

```
         +--:(dscp-marking)
         |  +--rw dscp
         |        ...
         +--:(meter-inline)
         |  +--rw (meter-types)?
         |        ...
         +--:(meter-reference)
         |  +--rw meter
         |        ...
         +--:(traffic-group-marking)
         |  +--rw traffic-group
         |        ...
         +--:(child-policy) {qos-types:child-policy}?
         |  +--rw child-policy {qos-types:child-policy}?
         |        ...
         +--:(count) {qos-types:count}?
         |  +--rw count {qos-types:count}?
         |        ...
         +--:(named-count) {qos-types:named-counter}?
         |  +--rw named-counter {qos-types:named-counter}?
         |        ...
         +--:(queue-inline)
         |  +--rw queue
         |        ...
         +--:(scheduler-inline)
            +--rw scheduler
                  ...
```

Figure 6: ietf-diffserv abridged tree diagram

```
<CODE BEGINS>
file "ietf-diffserv@2024-01-31.yang"
module ietf-diffserv {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-diffserv";
  prefix diffserv;

  import ietf-traffic-policy {
    prefix traffic-policy;
  }
  import ietf-qos-action {
    prefix qos-action;
  }
  import ietf-inet-types {
    prefix inet;
  }
  import iana-qos-types {
    prefix qos-types;
```

```
      reference
        "RFC XXXX: YANG Models for Quality of Service (QoS).";
    }

    organization
      "IETF Routing Area Working Group";

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

       Editor:   Aseem Choudhary
                 <mailto:achoudhary@aviatrix.com>
       Editor:   Mahesh Jethanandani
                 <mailto:mjethanandani@gmail.com>";

    description
      "This module contains a collection of YANG definitions for
       configuring DiffServ specification implementations.

       Copyright (c) 2023 IETF Trust and the persons identified as
       authors of the code.  All rights reserved.

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

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

    revision 2024-01-31 {
      description
        "Initial version.";
      reference
        "RFC XXXX: A YANG Model for Quality of Service (QoS).";
    }

    grouping dscp {
      list dscp {
        key "min max";
        description
          "List of DSCP ranges.";
        leaf min {
          type inet:dscp;
          description
```

```
            "Minimum value of DSCP min-max range.";
        }
        leaf max {
          type inet:dscp;
          must ". >= ../min" {
            error-message
            "max must be greater than or equal to min.";
          }
          description
            "Maximum value of DSCP min-max range.";
        }
      }
      description
        "Filter grouping containing list of DSCP ranges.";
    }

    grouping source-ipv4-prefix {
      list source-ipv4-prefix {
        key "prefix";
        description
          "List of source IPv4 prefixes.";
        leaf prefix {
          type inet:ipv4-prefix;
          description
            "Source IPv4 prefix.";
        }
      }
      description
        "Filter grouping containing list of source IPv4 prefixes.";
    }

    grouping destination-ipv4-prefix {
      list destination-ipv4-prefix {
        key "prefix";
        description
          "List of destination IPv4 prefixes.";
        leaf prefix {
          type inet:ipv4-prefix;
          description
            "Destination IPv4 prefix.";
        }
      }
      description
        "Filter grouping containing list of destination IPv4
         prefixes.";
    }

    grouping source-ipv6-prefix {
```

```
      list source-ipv6-prefix {
        key "prefix";
        description
          "List of source IPv6 prefixes.";
        leaf prefix {
          type inet:ipv6-prefix;
          description
            "Source IPv6 prefix.";
        }
      }
      description
        "Filter grouping containing list of source IPv6
         prefixes.";
    }

    grouping destination-ipv6-prefix {
      list destination-ipv6-prefix {
        key "prefix";
        description
          "List of destination IPv6 prefixes.";
        leaf prefix {
          type inet:ipv6-prefix;
          description
            "Destination IPv6 prefix.";
        }
      }
      description
        "Filter grouping containing list of destination IPv6
         prefixes.";
    }

    grouping source-port {
      list source-port {
        key "min max";
        description
          "List of ranges of source port.";

        leaf min {
          type inet:port-number;
          description
            "Minimum value of source port range.";
        }

        leaf max {
          type inet:port-number;
          must ". >= ../min" {
            error-message
            "max must be greater than or equal to min.";
```

```
          }
          description
            "Maximum value of source port range.";
        }
      }
      description
        "Filter grouping containing list of source port ranges.";
    }

    grouping destination-port {
      list destination-port {
        key "min max";
        description
          "List of min-max range of destination ports.";
        leaf min {
          type inet:port-number;
          description
            "Minimum value of destination port.";
        }
        leaf max {
          type inet:port-number;
          must ". >= ../min" {
            error-message
            "max must be greater than or equal to min.";
          }
          description
            "Maximum value of destination port range.";
        }
      }
      description
        "Filter grouping containing list of destination port ranges.";
    }

    grouping protocol {
      list protocol {
        key "min max";
        description
          "List of ranges of protocol values. Protocol refers to the
           value in the protocol field of the IPv4 header and value
           in the 'next-header' field of IPv6 header. In IPv6 header,
           'next-header' field indicates first extension header or the
            protocol in the 'upper-layer' header.";
        reference
          "RFC 791: Internet Protocol
           RFC 8200: Internet Protocol, Version 6 (IPv6)
                     Specification.";

        leaf min {
```

```
          type uint8 {
            range "0..255";
          }
          description
            "Minimum value of protocol range.";
        }

        leaf max {
          type uint8 {
            range "0..255";
          }
          must ". >= ../min" {
            error-message
            "max must be greater than or equal to min.";
          }
          description
            "Maximum value of protocol range.";
        }
      }
      description
        "Filter grouping containing list of protocol ranges.";
    }

    grouping traffic-group {
      container traffic-group {
        leaf name {
          type string ;
          description
            "This leaf defines name of the traffic group referenced.";
        }
      description
        "Traffic group container.";
      }
      description
        "Traffic group grouping.";
    }

    grouping filter-match-all {
      container match-all {
        leaf action {
          type empty;
          description
            "When set, the filter matches all packets.";
        }
        description
          "The match all action.";
      }
      description
```

```
        "The match all filter grouping.";
    }

    augment "/traffic-policy:classifiers/traffic-policy:classifier" +
            "/traffic-policy:filter" {
      choice filter-param {
        description
          "Choice of filter types.";
        case dscp {
          uses dscp;
          description
            "Filter containing list of DSCP ranges.";
        }

        case source-ipv4-prefix {
          uses source-ipv4-prefix;
          description
            "Filter containing list of source IPv4 prefixes.";
        }

        case destination-ipv4-prefix {
          uses destination-ipv4-prefix;
          description
            "Filter containing list of destination IPv4 prefix.";
        }

        case source-ipv6-prefix {
          uses source-ipv6-prefix;
          description
            "Filter containing list of source IPv6 prefixes.";
        }

        case destination-ipv6-prefix {
          uses destination-ipv6-prefix;
          description
            "Filter containing list of destination IPv6 prefix.";
        }

        case source-port {
          uses source-port;
          description
            "Filter containing list of source-port ranges.";
        }

        case destination-port {
          uses destination-port;
          description
            "Filter containing list of destination-port ranges.";
```

```
        }

        case protocol {
          uses protocol;
          description
            "Filter Type Protocol.";
        }

        case traffic-group {
          uses traffic-group;
          description
            "Filter Type traffic-group.";
        }

        case filter-match-all {
          uses filter-match-all;
          description
            "When the filter type is set to match all packets.";
        }
      }
      description
        "Augments QoS classifier to add in DiffServ filters.";
    }

    augment "/traffic-policy:policies/traffic-policy:policy" +
            "/traffic-policy:classifier/traffic-policy:inline" +
            "/traffic-policy:filter" {
      when "derived-from-or-self(../../../traffic-policy:type, " +
           "'qos-types:ipv4-diffserv-policy-type') or " +
           "derived-from-or-self(../../../traffic-policy:type, " +
           "'qos-types:ipv6-diffserv-policy-type') or " +
           "derived-from-or-self(../../../traffic-policy:type, " +
           "'qos-types:diffserv-policy-type')" {
        description
          "If policy type is v4, v6 or default DiffServ,
           this filter can be used.";
      }
      choice filter-params {
        description
          "Choice of action types.";
        case dscp {
          uses dscp;
          description
            "Filter containing list of DSCP ranges.";
        }

        case source-ipv4-prefix {
          when "derived-from-or-self(/traffic-policy:policies/" +
```

```
                "traffic-policy:policy/traffic-policy:type, " +
                "'qos-types:ipv6-diffserv-policy-type')" {
              description
                 "If policy type is v6, this filter cannot be used.";
            }
            uses source-ipv4-prefix;
            description
              "Filter containing list of source IPv4 prefixes.";
          }

          case destination-ipv4-prefix {
            when "derived-from-or-self(/traffic-policy:policies" +
                 "/traffic-policy:policy/traffic-policy:type, " +
                 "'qos-types:ipv6-diffserv-policy-type')" {
              description
                 "If policy type is v6, this filter cannot be used.";
            }
            uses destination-ipv4-prefix;
            description
              "Filter containing list of destination IPv4 prefix.";
          }

          case source-ipv6-prefix {
            when "derived-from-or-self(/traffic-policy:policies" +
                 "/traffic-policy:policy/traffic-policy:type, "+
                 "'qos-types:ipv4-diffserv-policy-type')" {
              description
                 "If policy type is v4, this filter cannot be used.";
            }
            uses source-ipv6-prefix;
            description
              "Filter containing list of source IPv6 prefixes.";
          }

          case destination-ipv6-prefix {
            when "derived-from-or-self(/traffic-policy:policies" +
                 "/traffic-policy:policy/traffic-policy:type, " +
                 "'qos-types:ipv4-diffserv-policy-type')" {
              description
                 "If policy type is v4, this filter cannot be used.";
            }
            uses destination-ipv6-prefix;
            description
              "Filter containing list of destination IPv6 prefix.";
          }

          case source-port {
            uses source-port;
```

```
          description
            "Filter containing list of source-port ranges.";
        }

        case destination-port {
          uses destination-port;
          description
            "Filter containing list of destination-port ranges.";
        }

        case protocol {
          uses protocol;
          description
            "Filter Type Protocol.";
        }

        case traffic-group {
          uses traffic-group;
          description
            "Filter Type traffic-group.";
        }

        case filter-match-all {
          uses filter-match-all;
          description
            "When the filter type is set to match all packets.";
        }
      }
      description
        "Augments Diffserv Classifier with common filter types.";
    }

    augment "/traffic-policy:policies/traffic-policy:policy/" +
            "traffic-policy:classifier/traffic-policy:action" {
      when "derived-from-or-self(../../traffic-policy:type, " +
           "'qos-types:ipv4-diffserv-policy-type') or " +
           "derived-from-or-self(../../traffic-policy:type, " +
           "'qos-types:ipv6-diffserv-policy-type') or " +
           "derived-from-or-self(../../traffic-policy:type, " +
           "'qos-types:diffserv-policy-type')" {
        description
          "If policy type is v4, v6 or default DiffServ,
           these actions can be used.";
      }
      description
        "Augments Diffserv Policy with action configuration.";

      choice action-params {
```

```
        description
          "Choice of different action parameters that can be taken.";

        case dscp-marking {
          uses qos-action:dscp-marking;
        }

        case meter-inline {
          uses qos-action:meter;
        }

        case meter-reference {
          uses qos-action:meter-reference;
        }

        case traffic-group-marking {
          uses qos-action:traffic-group-marking;
        }

        case child-policy {
          if-feature qos-types:child-policy;
          uses qos-action:child-policy;
        }

        case count {
          if-feature qos-types:count;
          uses qos-action:count;
        }

        case named-count {
          if-feature qos-types:named-counter;
          uses qos-action:named-counter;
        }

        case queue-inline {
          uses qos-action:queue;
        }

        case scheduler-inline {
          uses qos-action:scheduler;
        }
      }
    }
  }
  <CODE ENDS>
```

Figure 7: ietf-diffserv module

4.4. QoS Operational Module

QoS Operational module contains all operational statistics. It contains statistics related to classifier, metering, queueing, and named.

Classifier statistics consist of list of classifier entries identified by a classifier entry name. Classifier counters include matched packets and bytes, and average rate of traffic matching a particular classifier.

Metering statistics consist of meters identified by an identifier. Metering counters include conform, exceed, violate, drop packets, and bytes.

Queuing counters include instantaneous, peak, average queue length, as well as output conform, exceed, tail drop packets and bytes.

Named statistics is defined as statistics which is tagged by a name. This could be aggregated or non-aggregated. Aggregated named statistics is defined as counters which are aggregated across classifier entries in a policy applied to an interface in a particular direction. Non-aggregated named statistics are counters of classifier, metering or queuing which have the same tag name but maintained separately.

A clear action is provided to clear all statistics or statistics of a particular kind.

```
module: ietf-qos-oper

  augment /if:interfaces/if:interface:
    +--ro qos-interface-statistics
       +--ro stats-per-direction* []
       |  +--ro direction?     identityref
       |  +--ro policy-name?   string
       |  +--ro classifier* []
       |  |     ...
       |  +--ro named* []
       |  |     ...
       |  +--ro metering* []
       |  |     ...
       |  +--ro queueing* []
       |        ...
       +---x clear
          +---w input
          |     ...
          +--ro output
                ...
```

Figure 8: ietf-qos-oper abridged tree diagram

```
<CODE BEGINS>
file "ietf-qos-oper@2024-01-31.yang"
module ietf-qos-oper {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-qos-oper";
  prefix qos-oper;

  import ietf-yang-types {
    prefix yang;
    reference
      "RFC 6991:Common YANG Data Types.";
  }
  import ietf-interfaces {
    prefix if;
    reference
      "RFC8343: A YANG Data Model for Interface Management.";
  }
  import iana-qos-types {
    prefix qos-types;
    reference
      "RFC XXXX: YANG Models for Quality of Service (QoS).";
  }


  organization "IETF RTG (Routing Area) Working Group.";
```

```
     contact
       "WG Web:   <http://tools.ietf.org/wg/rtgwg/>
        WG List:  <mailto:rtgwg@ietf.org>
        Editor:   Aseem Choudhary
                  <mailto:achoudhary@aviatrix.com>
        Editor:   Mahesh Jethanandani
                  <mailto:mjethanandani@gmail.com>";
     description
       "This module contains a collection of YANG definitions for
        QoS operational specification.

        Copyright (c) 2023 IETF Trust and the persons identified as
        authors of the code.  All rights reserved.

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

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

     revision 2024-01-31 {
       description
         "Initial version.";
       reference
         "RFC XXXX: YANG Models for Quality of Service (QoS).";
     }

     grouping classifier-statistics {
       description
         "Statistics for a classifier.";

       leaf name {
         type string;
         description
           "A unique identifier for Classifier name.";
       }

       leaf classified-pkts {
         type yang:zero-based-counter64;
         description
           "Number of total packets filtered for a classifier.";
       }

       leaf classified-bytes {
```

```
        type yang:zero-based-counter64;
        description
          "Number of total bytes filtered for a classifier.";
      }

      leaf classified-rate {
        type yang:gauge64;
        units "bits-per-second";
        description
          "Rate of average data flow through a classifier.";
      }
    }

    grouping queuing-stats {
      description
        "Statistics for a queue.";

      leaf queue-id {
        type string;
        description
          "A unique identifier for the Queue.";
      }

      leaf transmit-pkts {
        type yang:zero-based-counter64;
        description
          "Number of packets transmitted from queue.";
      }

      leaf transmit-bytes {
        type yang:zero-based-counter64;
        description
          "Number of bytes transmitted from queue.";
      }

      leaf queue-current-size-bytes {
        type yang:gauge64;
        description
          "Number of bytes currently buffered.";
      }

      leaf queue-average-size-bytes {
        type yang:gauge64;
        description
          "Average queue size in number of bytes.";
      }

      leaf queue-peak-size-bytes {
```

```
        type yang:gauge64;
        description
          "Peak buffer queue size in bytes.";
      }

      leaf tail-drop-pkts {
        type yang:zero-based-counter64;
        description
          "Total number of packets tail-dropped.";
      }

      leaf tail-drop-bytes {
        type yang:zero-based-counter64;
        description
          "Total number of bytes tail-dropped.";
      }

      leaf red-drop-pkts {
        type yang:zero-based-counter64;
        description
          "Total number of packets dropped because of RED.";
      }

      leaf red-drop-bytes {
        type yang:zero-based-counter64;
        description
          "Total number of bytes dropped because of RED.";
      }

      leaf red-ecn-marked-pkts {
        type yang:zero-based-counter64;
        description
          "Total number of packets that were marked with ECN
           because of RED.";
      }

      leaf red-ecn-marked-bytes {
        type yang:zero-based-counter64;
        description
          "Total number of bytes that were marked with ECN because of
           RED.";
      }

      list wred-stats {
        config false;
        description
          "QoS WRED statistics.";
```

```
        leaf profile {
          type uint8;
          description
            "Profile identifier for each color of traffic.";
        }

        leaf drop-pkts {
          type yang:zero-based-counter64;
          description
            "Total number of packets dropped because of WRED.";
        }

        leaf drop-bytes {
          type yang:zero-based-counter64;
          description
            "Total number of bytes dropped because of WRED.";
        }

        leaf ecn-marked-pkts {
          type yang:zero-based-counter64;
          description
            "Total number of packets that were marked with ECN because
             of WRED.";
        }

        leaf ecn-marked-bytes {
          type yang:zero-based-counter64;
          description
            "Total number of bytes that were marked with ECN because of
             WRED.";
        }
      }
    }

    grouping metering-stats {
      description
        "Statistics for a meter.";

      leaf meter-id {
        type string;
        description
          "A unique identifier that identifies this set of metering
           statistics.";
      }

      leaf conform-pkts {
        type yang:zero-based-counter64;
        description
```

```
          "Number of packets that conform to the rate and burst.";
      }

      leaf conform-bytes {
        type yang:zero-based-counter64;
        description
          "Total bytes count that conform to the rate and burst.";
      }

      leaf conform-rate {
        type yang:gauge64;
        units "bits-per-second";
        description
          "Traffic Rate measured as conformimg.";
      }

      leaf exceed-pkts {
        type yang:zero-based-counter64;
        description
          "Number of packets that exceed the peak rate and peak
           burst.";
      }

      leaf exceed-bytes {
        type yang:zero-based-counter64;
        description
          "Total number of bytes which exceed the peak rate and peak
           burst.";
      }

      leaf exceed-rate {
        type yang:gauge64;
        units "bits-per-second";
        description
          "Traffic Rate measured as exceeding.";
      }

      leaf violate-pkts {
        type yang:zero-based-counter64;
        description
          "Number of packets that were beyond peak rate
           and peak burst.";
      }

      leaf violate-bytes {
        type yang:zero-based-counter64;
        description
          "Total number of bytes which were beyond peak rate and
```

```
            peak burst.";
      }

      leaf violate-rate {
        type yang:gauge64;
        units "bits-per-second";
        description
          "Traffic Rate measured as violating.";
      }

      leaf drop-pkts {
        type yang:zero-based-counter64;
        description
          "Number of packets dropped by meter.";
      }

      leaf drop-bytes {
        type yang:zero-based-counter64;
        description
          "Number of bytes dropped by meter.";
      }
    }

    augment "/if:interfaces/if:interface" {
      description
        "Augments interface module to add QoS Target entry.";

      container qos-interface-statistics {
        config false;
        description
          "QoS Interface statistics.";

        list stats-per-direction {
          description
            "QoS interface statistics in ingress or egress direction.";

          leaf direction {
            type identityref {
              base qos-types:direction;
            }
            description
              "Direction of the traffic flow. Either ingress or
               egress.";
          }

          leaf policy-name {
            type string;
            description
```

```
               "Policy name for single level policy as well as
                for Hierarchical policies. For Hierarchical policies,
                this represent relative path as well as the last level
                policy name.";
           }

           list classifier {
             description
               "Statistics for each Classifier in a
                Policy applied in a particular direction.";
             uses classifier-statistics;
           }

           list named {
             description
               "Statistics for a statistics-name.";

             leaf name {
               type string;
               description
                 "A name for statistics.";
             }

             container aggregated {
               description
                 "Matched aggregated statistics for a statistics-name.";

               leaf pkts {
                 type yang:zero-based-counter64;
                 description
                   "Number of total matched packets associated
                    with a statistics name.";
               }

               leaf bytes {
                 type yang:zero-based-counter64;
                 description
                   "Number of total matched bytes associated
                    with a statistics name.";
               }

               leaf rate {
                 type yang:gauge64;
                 units "bits-per-second";
                 description
                   "Rate of average matched data which is associated
                    to a statistics name.";
               }
```

```
            }

            container non-aggregated {
              description
                "Statistics for non-aggregated statistics-name.";

              list classifier {
                description
                  "Statistics for each Classifier in a
                   Policy applied in a particular direction.";
                uses classifier-statistics;
              }

              list metering {
                description
                  "Statistics for each Meter associated with
                   the Policy.";
                reference
                  "RFC2697: A Single Rate Three Color Marker
                   RFC2698: A Two Rate Three Color Marker.";
                uses metering-stats;
              }

              list queueing {
                description
                  "Statistics for each Queue associated with
                   the Policy.";
                uses queuing-stats;
              }
            }
          }

          list metering {
            description
              "Statistics for each Meter associated with the Policy.";
            reference
              "RFC2697: A Single Rate Three Color Marker
               RFC2698: A Two Rate Three Color Marker.";
            uses metering-stats;
          }

          list queueing {
            description
              "Statistics for each Queue associated with the Policy.";
            uses queuing-stats;
          }
        }
```

```
         action clear {
           description
             "Clear statistics action command. Execution of this command
              should result in all the QoS interface counters to be
              cleared and set to 0.";

           input {
             leaf category {
               type identityref {
                 base qos-types:clear-counters-type;
               }
               description
                 "Category of counters which is being cleared.";
             }

             leaf started-at {
               type yang:date-and-time;
               description
                 "The date and time when the counters should be
                  cleared.";
             }
           }

           output {
             leaf finished-at {
               type yang:date-and-time;
               description
                 "The date and time when the counters finished
                  clearing.";
             }
           }
         }
       }
     }
   }
   <CODE ENDS>
```

Figure 9: ietf-qos-oper module

4.5. Scheduler Policy Module

Scheduling Policy module defines a set of scheduling parameters and associates a Queue Policy with it.

```
module: ietf-scheduler-policy

  augment /traffic-policy:policies/traffic-policy:policy
            /traffic-policy:classifier/traffic-policy:inline
            /traffic-policy:filter:
    +--rw (filter-params)?
       +--:(filter-match-all)
          +--rw match-all
                ...
  augment /traffic-policy:policies/traffic-policy:policy
            /traffic-policy:classifier/traffic-policy:action:
    +--rw (action-params)?
       +--:(scheduler)
       |  +--rw scheduler
       |        ...
       +--:(queue-policy-name)
          +--rw queue-policy-name
                ...
```

Figure 10: ietf-scheduler-policy abridged abridged tree diagram

```
<CODE BEGINS>
file "ietf-scheduler-policy@2024-01-31.yang"
module ietf-scheduler-policy {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-scheduler-policy";
  prefix scheduler-policy;

  import ietf-traffic-policy {
    prefix traffic-policy;
  }
  import ietf-qos-action {
    prefix qos-action;
  }
  import ietf-diffserv {
    prefix diffserv;
  }
  import iana-qos-types {
    prefix qos-types;
    reference
      "RFC XXXX: YANG Models for Quality of Service (QoS).";
  }

  organization
    "IETF Routing Area Working Group.";

  contact
    "WG Web:   <https://datatracker.ietf.org/wg/rtgwg/>
```

```
        WG List:  <mailto:rtgwg@ietf.org>

        Editor:   Aseem Choudhary
                  <mailto:achoudhary@aviatrix.com>
        Editor:   Mahesh Jethanandani
                  <mailto:mjethanandani@gmail.com>";

     description
       "This module contains a collection of YANG definitions for
        configuring QoS scheduler policies.

        Copyright (c) 2023 IETF Trust and the persons identified as
        authors of the code.  All rights reserved.

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

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

     revision 2024-01-31 {
       description
         "Initial version.";
       reference
         "RFC XXXX: YANG Models for Quality of Service (QoS).";
     }

     augment "/traffic-policy:policies/traffic-policy:policy" +
             "/traffic-policy:classifier/traffic-policy:inline" +
             "/traffic-policy:filter" {
       when "derived-from-or-self(../../../traffic-policy:type, " +
            "'qos-types:scheduler-policy-type')" {
         description
           "This filter can be used only when policy type is
            scheduler-policy.";
       }
       choice filter-params {
         description
           "Choice of filter parameters.";
         case filter-match-all {
           uses diffserv:filter-match-all;
           description
             "A filter that matches all packets.";
         }
```

```
      }
      description
        "Augments Queue Policy Classifier to add filter parameters.";
    }

    augment "/traffic-policy:policies/traffic-policy:policy/" +
            "traffic-policy:classifier/traffic-policy:action" {
      when "derived-from-or-self(../../traffic-policy:type, " +
           "'qos-types:scheduler-policy-type')" {
        description
          "These actions apply only when policy type is
           scheduler-policy.";
      }

      choice action-params {
        description
          "Choice of action parameters that can be taken.";

        case scheduler {
          uses qos-action:scheduler;
        }

        case queue-policy-name {
          container queue-policy-name {
            leaf queue-policy {
              type leafref  {
                path "/traffic-policy:policies/" +
                     "traffic-policy:policy/traffic-policy:name";
              }
              must "derived-from-or-self(/traffic-policy:policies" +
                   "/traffic-policy:policy/traffic-policy:type," +
                   "'qos-types:queue-policy-type')";
              mandatory true;
              description
                "This leafref refers to Queue Policy name.";
            }
            description
              "Container for Queue Policy name.";
          }
        }
      }
      description
        "Augments Queue Policy Classifier to add in Action parameters
         for Scheduler policy.";
    }
  }
  <CODE ENDS>
```

Figure 11: ietf-scheduler-policy module

4.6. Queue Policy Module

Queue Policy module augments the Traffic Policy module. It augments to allow queues to be referred to in a policy. The match is based on Traffic group and action parameters which are defined in QoS Action module.

```
module: ietf-queue-policy

  augment /traffic-policy:policies/traffic-policy:policy
            /traffic-policy:classifier/traffic-policy:inline
            /traffic-policy:filter:
    +--rw (filter-params)?
       +--:(traffic-group)
       |  +--rw traffic-group
       |        ...
       +--:(filter-match-all)
          +--rw match-all
                ...
  augment /traffic-policy:policies/traffic-policy:policy
            /traffic-policy:classifier/traffic-policy:action:
    +--rw (action-params)?
       +--:(queue-template-name)
       |  +--rw queue-reference
       |        ...
       +--:(queue-inline)
          +--rw queue
                ...
```

Figure 12: ietf-queue-policy abridged tree diagram

```
<CODE BEGINS>
file "ietf-queue-policy@2024-01-31.yang"
module ietf-queue-policy {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-queue-policy";
  prefix queue-policy;

  import ietf-traffic-policy {
    prefix traffic-policy;
  }
  import ietf-qos-action {
    prefix qos-action;
  }
  import ietf-diffserv {
    prefix diffserv;
```

```
     }
     import iana-qos-types {
       prefix qos-types;
       reference
         "RFC XXXX: YANG Models for Quality of Service (QoS).";
     }

     organization
       "IETF Routing Area Working Group.";

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

        Editor:   Aseem Choudhary
                  <mailto:achoudhary@aviatrix.com>
        Editor:   Mahesh Jethanandani
                  <mailto:mjethanandani@gmail.com>";

     description
       "This module contains a collection of YANG definitions for
        QoS Queue policies.

        Copyright (c) 2023 IETF Trust and the persons identified as
        authors of the code.  All rights reserved.

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

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

     revision 2024-01-31 {
       description
         "Initial version.";
       reference
         "RFC XXXX: YANG Models for Quality of Service (QoS).";
     }

     augment "/traffic-policy:policies/traffic-policy:policy" +
             "/traffic-policy:classifier/traffic-policy:inline" +
             "/traffic-policy:filter" {
       when "derived-from-or-self(../../../traffic-policy:type, " +
            "'qos-types:queue-policy-type')" {
```

```
        description
          "If policy type is Queue Policy, this filter can be used.";
      }

      choice filter-params {
        description
          "Choice of Action types.";
        case traffic-group {
          uses diffserv:traffic-group;
          description
            "Traffic group name.";
        }
        case filter-match-all {
          uses diffserv:filter-match-all;
          description
            "When the filter type is set to match all packets.";
        }
      }
      description
        "Augments Queue Policy to add filter types.";
    }

    augment "/traffic-policy:policies/traffic-policy:policy" +
            "/traffic-policy:classifier/traffic-policy:action" {
      when "derived-from-or-self(../../traffic-policy:type,"  +
           "'qos-types:queue-policy-type')" {
        description
          "Queue Policy actions.";
      }

      choice action-params {
        description
          "Choice of different action parameters that can be taken.";

        case queue-template-name {
          uses qos-action:queue-reference;
        }

        case queue-inline {
          uses qos-action:queue;
        }
      }
      description
        "Augments Queue Policy action to add Queue template reference
         or Queue inline.";
    }
  }
  <CODE ENDS>
```

Figure 13: ietf-queue-policy module

4.7. IANA QoS Types Module

The QoS Types module contains all the type definitions used by the other QoS modules.

```
<CODE BEGINS>
file "iana-qos-types@2024-01-31.yang"
module iana-qos-types {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:iana-qos-types";
  prefix qos-types;

  organization
    "IETF Routing Area Working Group";

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

     Editor:   Aseem Choudhary
               <mailto:asechoud@cisco.com>
     Editor:   Mahesh Jethanandani
               <mailto:mjethanandani@gmail.com>";

  description
    "This module contains type definitions for all QoS types.

     Copyright (c) 2022 IETF Trust and the persons identified as
     authors of the code.  All rights reserved.

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

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

  revision 2024-01-31 {
    description
      "Initial version";
    reference
      "RFC XXXX: YANG Models for Quality of Service (QoS).";
  }
```

```
     /*
      * Features.
      */
     feature child-policy {
       description
         " This feature allows configuration of hierarchical policy.";
     }
     feature count {
       description
         "This feature allows action configuration to enable
          counter in a classifier";
     }
     feature named-counter {
       description
         "This feature allows action configuration to enable
          named counter in a classifier";
     }

     /*
      * Identities.
      */
     identity policy-type {
       description
         "Base identity for policy type.";
     }
     identity diffserv-policy-type {
       base policy-type;
       description
         "Policy type defined as DiffServ.";
     }
     identity ipv4-diffserv-policy-type {
       base policy-type;
       description
         "Policy type defined as a DiffServ IPv4 policy type.";
     }
     identity ipv6-diffserv-policy-type {
       base policy-type;
       description
         "Policy type defined as a DiffServ IPv6 policy type.";
     }
     identity queue-policy-type {
       base policy-type;
       description
         "Policy type defined as a queue policy type.";
     }
     identity scheduler-policy-type {
       base policy-type;
       description
```

```
        "Policy type defined as a scheduler policy type.";
    }

    identity action-type {
      description
        "Base identity for the action type.";
    }
    identity dscp-marking {
      base action-type;
      description
        "Action type defined as DSCP marking.";
    }
    identity meter-inline {
      base action-type;
      description
        "Action type defined as meter inline.";
    }
    identity meter-reference {
      base action-type;
      description
        "Action type defined as meter reference.";
    }
    identity queue {
      base action-type;
      description
        "Action type defined as queue.";
    }
    identity scheduler {
      base action-type;
      description
        "Action type defined as scheduler.";
    }
    identity discard {
      base action-type;
      description
        "Action type defined as discard.";
    }
    identity child-policy {
      if-feature child-policy;
      base action-type;
      description
        "Action type defined as child policy.";
    }
    identity count {
      if-feature count;
      base action-type;
      description
        "Action type defined as count.";
```

```
     }
     identity named-counter {
       if-feature named-counter;
       base action-type;
       description
         "Action type specified as a named counter.";
     }
     identity queue-policy-name {
       base action-type;
       description
         "Action type specified as a queue policy name.";
     }

     identity meter-type {
       description
         "Base identity for types of meter.";
     }
     identity one-rate-two-color {
       base meter-type;
       description
         "Meter type specified as one rate two color meter.";
     }
     identity one-rate-tri-color {
       base meter-type;
       description
         "Meter type specified as one rate three color meter.";
     }
     identity two-rate-tri-color {
       base meter-type;
       description
         "Meter type specified as two rate three color meter.";
     }

     identity drop-type {
       description
         "Base identity for drop algorithm.";
     }
     identity tail-drop {
       base drop-type;
       description
         "Drop algorithm specified as tail drop.";
     }
     identity red {
       base drop-type;
       description
         "Drop algorithm specified as RED.";
     }
     identity wred {
```

```
      base drop-type;
      description
        "Drop algorithm specified as WRED.";
    }

    identity rate-unit-type {
      description
        "Base definition for the type of rate unit.";
    }
    identity bits-per-second {
      base rate-unit-type;
      description
        "Rate specified in bits per second.";
    }
    identity percent {
      base rate-unit-type;
      description
        "Rate specified as a percentage of queue depth.";
    }

    identity burst-unit-type {
      description
        "Base definition for burst specification.";
    }
    identity bytes {
      base burst-unit-type;
      description
        "Burst specified in bytes.";
    }
    identity microsecond {
      base burst-unit-type;
      description
        "Burst specified in microseconds.";
    }

    identity red-threshold-unit {
      description
        "Base definition for RED threshold specification.";
    }
    identity red-threshold-bytes {
      base red-threshold-unit;
      description
        "RED threshold specified in bytes.";
    }
    identity red-threshold-us {
      base red-threshold-unit;
      description
        "RED threshold specified in microseconds.";
```

```
     }
     identity red-threshold-pc {
       base red-threshold-unit;
       description
         "RED threshold specified as a percentage.";
     }
     identity red-theshold-pt {
       base red-threshold-unit;
       description
         "RED threshold specified per-thousand.";
     }

     identity wred-color-type {
       description
         "Base identity for WRED color type.";
     }
     identity wred-color-dscp {
       base wred-color-type;
       description
         "WRED color specified by DSCP.";
     }

     identity probability-unit {
       description
         "Base definition for probability unit.";
     }
     identity probability-pc {
       base probability-unit;
       description
         "Probability specified in percentage.";
     }
     identity probability-pt {
       base probability-unit;
       description
         "Probability specified in unit of per thousand.";
     }
     identity probability-denominator {
       base probability-unit;
       description
         "Probability value specified as a denominator value
          while numerator is always 1.";
     }

     identity filter-type {
       description
         "Base identity for filter type.";
     }
```

```
     identity dscp {
       base filter-type;
       description
         "Filter type specified as DSCP.";
     }
     identity source-ipv4-prefix {
       base filter-type;
       description
         "Filter type specified as source IPv4 prefix.";
     }
     identity destination-ipv4-prefix {
       base filter-type;
       description
         "Filter type specified as destination IPv4 prefix.";
     }
     identity source-ipv6-prefix {
       base filter-type;
       description
         "Filter type specified as source IPv6 prefix.";
     }
     identity destination-ipv6-prefix {
       base filter-type;
       description
         "Filter type specified as destination IPv6 prefix.";
     }
     identity source-port {
       base filter-type;
       description
         "Filter type specified as source port.";
     }
     identity destination-port {
       base filter-type;
       description
         "Filter type specified as destination port.";
     }
     identity protocol {
       base filter-type;
       description
         "Filter type specified as protocol.";
     }
     identity traffic-group-name {
       base filter-type;
       description
         "Filter type specified as traffic group name.";
     }
     identity filter-match-all {
       base filter-type;
       description
```

```
        "Filter type specified as match all.";
    }

    typedef match-operation-type {
      type enumeration {
        enum match-all {
          description
            "Classifier entry filters logical AND operation";
        }
        enum match-any {
          description
            "Classifier entry filters logical OR operation";
        }
      }
      default "match-all";
      description
        "Filter match logical operation type";
    }

    identity direction {
      description
        "Base identity for traffic direction";
    }
    identity ingress {
      base direction;
      description
        "Direction of traffic coming into the network entry";
    }
    identity egress {
      base direction;
      description
        "Direction of traffic going out of the network entry";
    }

    identity meter-action-type {
      description
        "Base identify for actions related to metering";
    }
    identity action-drop {
      base meter-action-type;
        description
          "Drop packets that conform/exceed/violate to the set value.";
    }
    identity action-transmit {
      base meter-action-type;
        description
          "Transmit packets that conform/exceed/violate to the set
           value.";
```

```
     }
     identity action-mark {
       base meter-action-type;
         description
           "Mark packets that conform/exceed/violate to the set value.";
     }

     identity clear-counters-type {
       description
         "Base identify for clear interface counters related options";
     }
     identity all-counters {
       base clear-counters-type;
         description
           "Clear all counters in both directions.";
     }
     identity inbound-counters {
       base clear-counters-type;
         description
           "Clear counters in inbound direction.";
     }
     identity outbound-counters {
       base clear-counters-type;
         description
           "Clear counters in outbound direction.";
     }
   }
   <CODE ENDS>
```

Figure 14: iana-qos-types module

5. IANA Considerations

This document registers seven URIs and seven YANG modules.

5.1. URI Registration

Following the format in the IETF XML registry [RFC3688], the following registration is requested to be made:

```
URI: urn:ietf:params:xml:ns:yang:iana-qos-types
URI: urn:ietf:params:xml:ns:yang:ietf-diffserv
URI: urn:ietf:params:xml:ns:yang:ietf-qos-action
URI: urn:ietf:params:xml:ns:yang:ietf-qos-oper
URI: urn:ietf:params:xml:ns:yang:ietf-queue-policy
URI: urn:ietf:params:xml:ns:yang:ietf-scheduler-policy
URI: urn:ietf:params:xml:ns:yang:ietf-traffic-policy
```

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

5.2. YANG Module Name Registration

This document registers seven YANG modules in the YANG Module Names registry YANG [RFC6020].

```
name: iana-qos-types
namespace: urn:ietf:params:xml:ns:yang:iana-qos-types
prefix: qos-types
reference: RFC XXXX

name: ietf-diffserv
namespace: urn:ietf:params:xml:ns:yang:ietf-diffserv
prefix: diffserv
reference: RFC XXXX

name: ietf-qos-action
namespace: urn:ietf:params:xml:ns:yang:ietf-qos-action
prefix: action
reference: RFC XXXX

name: ietf-qos-oper
namespace: urn:ietf:params:xml:ns:yang:ietf-qos-oper
prefix: oper
reference: RFC XXXX

name: ietf-queue-policy
namespace: urn:ietf:params:xml:ns:yang:ietf-queue-policy
prefix: queue-policy
reference: RFC XXXX

name: ietf-scheduler-policy
namespace: urn:ietf:params:xml:ns:yang:ietf-scheduler-policy
prefix: scheduler-policy
reference: RFC XXXX

name: ietf-traffic-policy
namespace: urn:ietf:params:xml:ns:yang:ietf-traffic-policy
prefix: policy
reference: RFC XXXX
```

6. Security Considerations

The YANG module specified in this document defines a schema for data that is designed to be accessed via network management protocols such as NETCONF [RFC6241] or RESTCONF [RFC8040]. The lowest NETCONF layer is the secure transport layer, and the mandatory-to-implement secure transport is Secure Shell (SSH) [RFC6242]. The lowest RESTCONF layer is HTTPS, and the mandatory-to-implement secure transport is TLS [RFC8446]. The NETCONF Access Control Model (NACM) [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. Some of the subtrees and data nodes and their sensitivity/vulnerability are described here.

- 'filter-operation' and 'filter'. 'filter-operation' includes the ability to 'match all', which is a logical AND operation or 'match any', which is a logical OR operation. Both the operations have an impact on the traffic that is being classified. Similarly, the 'filter' list decides what parts of the packet will examined, which will also impact the traffic that is being classified.

- 'action'. The 'action' decides what action will be taken on the packet. That includes whether the packet will be marked, queued, or just discarded.

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

- 'conform-pkts', 'conform-bytes', 'exceed-pkts', 'exceed-bytes', 'voilate-pkts' or 'voilate-bytes'. All these statistics combined with their 'meter-id' are an indication of what is happening in the network, and can allow for an intruder insight into how to disrupt the traffic.

- 'tail-drop-pkts', 'red-drop-bytes', 'wred-stats' are examples of statistics that indicate the kind of traffic that is being profiled for drop or marked for Explicit Congestion Notification (ECN), and can give an insight into the operation of a network.

Some of the RPC operations in this YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control access to these operations. These are the operations and their sensitivity/vulnerability:

- The model allows for statistics to be cleared by the 'clear' RPC operation, causing all the individual statistics to be cleared.

7. Acknowledgement

The authors wish to thank Ruediger Geib, Fred Baker, Greg Mirsky, Tom Petch, Acee Lindem, many others for their helpful comments.

8. Contributors

The following people have substantially contributed to the editing of this document:

Norm Strahle
Email: nstrahle@juniper.net .

Shitanshu Shah
Email: svshah@cisco.com .

9. References

9.1. Normative References

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

[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>.

[RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, DOI 10.17487/RFC6991, July 2013, <https://www.rfc-editor.org/info/rfc6991>.

[RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <https://www.rfc-editor.org/info/rfc7950>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8342] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture (NMDA)", RFC 8342, DOI 10.17487/RFC8342, March 2018, <https://www.rfc-editor.org/info/rfc8342>.

[RFC8343] Bjorklund, M., "A YANG Data Model for Interface Management", RFC 8343, DOI 10.17487/RFC8343, March 2018, <https://www.rfc-editor.org/info/rfc8343>.

9.2. Informative References

[RFC2474] Nichols, K., Blake, S., Baker, F., and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, DOI 10.17487/RFC2474, December 1998, <https://www.rfc-editor.org/info/rfc2474>.

[RFC2475] Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z., and W. Weiss, "An Architecture for Differentiated Services", RFC 2475, DOI 10.17487/RFC2475, December 1998, <https://www.rfc-editor.org/info/rfc2475>.

[RFC2697] Heinanen, J. and R. Guerin, "A Single Rate Three Color Marker", RFC 2697, DOI 10.17487/RFC2697, September 1999, <https://www.rfc-editor.org/info/rfc2697>.

[RFC2698] Heinanen, J. and R. Guerin, "A Two Rate Three Color Marker", RFC 2698, DOI 10.17487/RFC2698, September 1999, <https://www.rfc-editor.org/info/rfc2698>.

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

[RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <https://www.rfc-editor.org/info/rfc6241>.

[RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <https://www.rfc-editor.org/info/rfc6242>.

[RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <https://www.rfc-editor.org/info/rfc8040>.

[RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, RFC 8340, DOI 10.17487/RFC8340, March 2018, <https://www.rfc-editor.org/info/rfc8340>.

[RFC8341] Bierman, A. and M. Bjorklund, "Network Configuration Access Control Model", STD 91, RFC 8341, DOI 10.17487/RFC8341, March 2018, <https://www.rfc-editor.org/info/rfc8341>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/info/rfc8446>.

Appendix A. Complete tree Diagrams

This section carries complete tree diagrams of each module defined in this draft. The tree diagrams use the notation defined in YANG Tree Diagrams [RFC8340].

A.1. Traffic Policy Module Tree Diagram

```
module: ietf-traffic-policy
  +--rw classifiers
  |  +--rw classifier* [name]
  |     +--rw name                string
  |     +--rw description?        string
  |     +--rw filter-operation?   qos-types:match-operation-type
  |     +--rw filter* [type logical-not]
  |        +--rw type           identityref
  |        +--rw logical-not    boolean
  +--rw policies
     +--rw policy* [name type]
        +--rw name           string
        +--rw type           identityref
        +--rw description?   string
        +--rw classifier* [name]
           +--rw name      string
           +--rw inline!
           |  +--rw filter-operation?
           |  |       qos-types:match-operation-type
           |  +--rw filter* [type logical-not]
           |     +--rw type           identityref
           |     +--rw logical-not    boolean
           +--rw action* [type]
              +--rw type    identityref

  augment /if:interfaces/if:interface:
    +--rw qos-target-policy* [direction type]
       +--rw direction    identityref
       +--rw type         identityref
       +--rw name         string
```

Figure 15: Traffic Policy Tree Diagram

A.2. QoS Action Module Tree Diagram

```
module: ietf-qos-action
  +--rw meters
  |  +--rw meter* [name]
  |     +--rw name                                    string
  |     +--rw (meter-types)?
  |        +--:(single-rate-two-color-meter-type)
  |        |  +--rw single-rate-two-color-meter
  |        |     +--rw committed-information-rate?   uint64
  |        |     +--rw committed-burst-size?         uint64
  |        |     +--rw conform-action
  |        |     |  +--rw action-type?          identityref
  |        |     |  +--rw dscp-mark?            inet:dscp
  |        |     |  +--rw traffic-group-mark?   string
```

```
     |         |      +--rw exceed-action
     |         |         +--rw action-type?          identityref
     |         |         +--rw dscp-mark?            inet:dscp
     |         |         +--rw traffic-group-mark?   string
     |         +--:(single-rate-three-color-meter-type)
     |         |  +--rw single-rate-three-color-meter
     |         |      +--rw committed-information-rate?   uint64
     |         |      +--rw committed-burst-size?         uint64
     |         |      +--rw excess-burst-size?            uint64
     |         |      +--rw conform-action
     |         |      |  +--rw action-type?          identityref
     |         |      |  +--rw dscp-mark?            inet:dscp
     |         |      |  +--rw traffic-group-mark?   string
     |         |      +--rw exceed-action
     |         |      |  +--rw action-type?          identityref
     |         |      |  +--rw dscp-mark?            inet:dscp
     |         |      |  +--rw traffic-group-mark?   string
     |         |      +--rw violate-action
     |         |         +--rw action-type?          identityref
     |         |         +--rw dscp-mark?            inet:dscp
     |         |         +--rw traffic-group-mark?   string
     |         +--:(two-rate-three-color-meter-type)
     |            +--rw two-rate-three-color-meter
     |               +--rw committed-information-rate?   uint64
     |               +--rw committed-burst-size?         uint64
     |               +--rw peak-information-rate?        uint64
     |               +--rw peak-burst-size?              uint64
     |               +--rw conform-action
     |               |  +--rw action-type?          identityref
     |               |  +--rw dscp-mark?            inet:dscp
     |               |  +--rw traffic-group-mark?   string
     |               +--rw exceed-action
     |               |  +--rw action-type?          identityref
     |               |  +--rw dscp-mark?            inet:dscp
     |               |  +--rw traffic-group-mark?   string
     |               +--rw violate-action
     |                  +--rw action-type?          identityref
     |                  +--rw dscp-mark?            inet:dscp
     |                  +--rw traffic-group-mark?   string
     +--rw queues
        +--rw queue* [name]
           +--rw name     string
           +--rw queue
              +--rw priority
              |  +--rw level?   uint8
              +--rw min-rate
              |  +--rw value?   uint64
              |  +--rw unit?    identityref
```

```
              +--rw max-rate
              |  +--rw value?         uint64
              |  +--rw unit?          identityref
              |  +--rw burst-value?   uint64
              |  +--rw burst-unit?    identityref
              +--rw algorithmic-drop
                 +--rw (drop-algorithm)?
                    +--:(tail-drop)
                    |  +--rw tail-drop
                    |     +--rw tail-drop?   empty
                    +--:(red)
                    |  +--rw red
                    |     +--rw min-threshold-val?      uint64
                    |     +--rw min-threshold-unit?     identityref
                    |     +--rw max-threshold-val?      uint64
                    |     +--rw max-threshold-unit?     identityref
                    |     +--rw weight?                 uint8
                    |     +--rw max-probability-val?    uint64
                    |     +--rw max-probability-unit?   identityref
                    |     +--rw ecn-enabled?            boolean
                    +--:(wred)
                       +--rw wred
                          +--rw wred* [profile]
                          |  +--rw profile                 uint8
                          |  +--rw color-type?             identityref
                          |  +--rw color-val* [min max]
                          |  |  +--rw min    uint8
                          |  |  +--rw max    uint8
                          |  +--rw min-threshold-val?      uint64
                          |  +--rw min-threshold-unit?     identityref
                          |  +--rw max-threshold-val?      uint64
                          |  +--rw max-threshold-unit?     identityref
                          |  +--rw weight?                 uint8
                          |  +--rw max-probability-val?    uint64
                          |  +--rw max-probability-unit?   identityref
                          +--rw ecn-enabled?   boolean
```

Figure 16: QoS Action Module Tree Diagram

A.3. DiffServ Module Tree Diagram

```
module: ietf-diffserv

  augment /traffic-policy:classifiers/traffic-policy:classifier
            /traffic-policy:filter:
    +--rw (filter-param)?
       +--:(dscp)
       |  +--rw dscp* [min max]
```

```
          |      +--rw min    inet:dscp
          |      +--rw max    inet:dscp
          +--:(source-ipv4-prefix)
          |  +--rw source-ipv4-prefix* [prefix]
          |      +--rw prefix    inet:ipv4-prefix
          +--:(destination-ipv4-prefix)
          |  +--rw destination-ipv4-prefix* [prefix]
          |      +--rw prefix    inet:ipv4-prefix
          +--:(source-ipv6-prefix)
          |  +--rw source-ipv6-prefix* [prefix]
          |      +--rw prefix    inet:ipv6-prefix
          +--:(destination-ipv6-prefix)
          |  +--rw destination-ipv6-prefix* [prefix]
          |      +--rw prefix    inet:ipv6-prefix
          +--:(source-port)
          |  +--rw source-port* [min max]
          |      +--rw min    inet:port-number
          |      +--rw max    inet:port-number
          +--:(destination-port)
          |  +--rw destination-port* [min max]
          |      +--rw min    inet:port-number
          |      +--rw max    inet:port-number
          +--:(protocol)
          |  +--rw protocol* [min max]
          |      +--rw min    uint8
          |      +--rw max    uint8
          +--:(traffic-group)
          |  +--rw traffic-group
          |      +--rw name?   string
          +--:(filter-match-all)
             +--rw match-all
                +--rw action?   empty
     augment /traffic-policy:policies/traffic-policy:policy
               /traffic-policy:classifier/traffic-policy:inline
               /traffic-policy:filter:
       +--rw (filter-params)?
          +--:(dscp)
          |  +--rw dscp* [min max]
          |      +--rw min    inet:dscp
          |      +--rw max    inet:dscp
          +--:(source-ipv4-prefix)
          |  +--rw source-ipv4-prefix* [prefix]
          |      +--rw prefix    inet:ipv4-prefix
          +--:(destination-ipv4-prefix)
          |  +--rw destination-ipv4-prefix* [prefix]
          |      +--rw prefix    inet:ipv4-prefix
          +--:(source-ipv6-prefix)
          |  +--rw source-ipv6-prefix* [prefix]
```

```
          |     +--rw prefix    inet:ipv6-prefix
          +--:(destination-ipv6-prefix)
          |  +--rw destination-ipv6-prefix* [prefix]
          |     +--rw prefix    inet:ipv6-prefix
          +--:(source-port)
          |  +--rw source-port* [min max]
          |     +--rw min    inet:port-number
          |     +--rw max    inet:port-number
          +--:(destination-port)
          |  +--rw destination-port* [min max]
          |     +--rw min    inet:port-number
          |     +--rw max    inet:port-number
          +--:(protocol)
          |  +--rw protocol* [min max]
          |     +--rw min    uint8
          |     +--rw max    uint8
          +--:(traffic-group)
          |  +--rw traffic-group
          |     +--rw name?   string
          +--:(filter-match-all)
             +--rw match-all
                +--rw action?   empty
     augment /traffic-policy:policies/traffic-policy:policy
               /traffic-policy:classifier/traffic-policy:action:
       +--rw (action-params)?
          +--:(dscp-marking)
          |  +--rw dscp
          |     +--rw dscp?   inet:dscp
          +--:(meter-inline)
          |  +--rw (meter-types)?
          |     +--:(single-rate-two-color-meter-type)
          |     |  +--rw single-rate-two-color-meter
          |     |     +--rw committed-information-rate?   uint64
          |     |     +--rw committed-burst-size?         uint64
          |     |     +--rw conform-action
          |     |     |  +--rw action-type?          identityref
          |     |     |  +--rw dscp-mark?            inet:dscp
          |     |     |  +--rw traffic-group-mark?   string
          |     |     +--rw exceed-action
          |     |        +--rw action-type?          identityref
          |     |        +--rw dscp-mark?            inet:dscp
          |     |        +--rw traffic-group-mark?   string
          |     +--:(single-rate-three-color-meter-type)
          |     |  +--rw single-rate-three-color-meter
          |     |     +--rw committed-information-rate?   uint64
          |     |     +--rw committed-burst-size?         uint64
          |     |     +--rw excess-burst-size?            uint64
          |     |     +--rw conform-action
```

```
          |     |     |  +--rw action-type?          identityref
          |     |     |  +--rw dscp-mark?            inet:dscp
          |     |     |  +--rw traffic-group-mark?   string
          |     |     +--rw exceed-action
          |     |     |  +--rw action-type?          identityref
          |     |     |  +--rw dscp-mark?            inet:dscp
          |     |     |  +--rw traffic-group-mark?   string
          |     |     +--rw violate-action
          |     |        +--rw action-type?          identityref
          |     |        +--rw dscp-mark?            inet:dscp
          |     |        +--rw traffic-group-mark?   string
          |     +--:(two-rate-three-color-meter-type)
          |        +--rw two-rate-three-color-meter
          |           +--rw committed-information-rate?   uint64
          |           +--rw committed-burst-size?         uint64
          |           +--rw peak-information-rate?        uint64
          |           +--rw peak-burst-size?              uint64
          |           +--rw conform-action
          |           |  +--rw action-type?          identityref
          |           |  +--rw dscp-mark?            inet:dscp
          |           |  +--rw traffic-group-mark?   string
          |           +--rw exceed-action
          |           |  +--rw action-type?          identityref
          |           |  +--rw dscp-mark?            inet:dscp
          |           |  +--rw traffic-group-mark?   string
          |           +--rw violate-action
          |              +--rw action-type?          identityref
          |              +--rw dscp-mark?            inet:dscp
          |              +--rw traffic-group-mark?   string
          +--:(meter-reference)
          |  +--rw meter
          |     +--rw name    string
          |     +--rw type    identityref
          +--:(traffic-group-marking)
          |  +--rw traffic-group
          |     +--rw traffic-group?   string
          +--:(child-policy) {qos-types:child-policy}?
          |  +--rw child-policy {qos-types:child-policy}?
          |     +--rw name?   string
          +--:(count) {qos-types:count}?
          |  +--rw count {qos-types:count}?
          |     +--rw count-action?   empty
          +--:(named-count) {qos-types:named-counter}?
          |  +--rw named-counter {qos-types:named-counter}?
          |     +--rw count-name-action?   string
          +--:(queue-inline)
          |  +--rw queue
          |     +--rw priority
```

```
          |      |  +--rw level?   uint8
          |      +--rw min-rate
          |      |  +--rw value?   uint64
          |      |  +--rw unit?    identityref
          |      +--rw max-rate
          |      |  +--rw value?         uint64
          |      |  +--rw unit?          identityref
          |      |  +--rw burst-value?   uint64
          |      |  +--rw burst-unit?    identityref
          |      +--rw algorithmic-drop
          |         +--rw (drop-algorithm)?
          |            +--:(tail-drop)
          |            |  +--rw tail-drop
          |            |     +--rw tail-drop?   empty
          |            +--:(red)
          |            |  +--rw red
          |            |     +--rw min-threshold-val?      uint64
          |            |     +--rw min-threshold-unit?     identityref
          |            |     +--rw max-threshold-val?      uint64
          |            |     +--rw max-threshold-unit?     identityref
          |            |     +--rw weight?                 uint8
          |            |     +--rw max-probability-val?    uint64
          |            |     +--rw max-probability-unit?   identityref
          |            |     +--rw ecn-enabled?            boolean
          |            +--:(wred)
          |               +--rw wred
          |                  +--rw wred* [profile]
          |                  |  +--rw profile                 uint8
          |                  |  +--rw color-type?             identityref
          |                  |  +--rw color-val* [min max]
          |                  |  |  +--rw min    uint8
          |                  |  |  +--rw max    uint8
          |                  |  +--rw min-threshold-val?      uint64
          |                  |  +--rw min-threshold-unit?     identityref
          |                  |  +--rw max-threshold-val?      uint64
          |                  |  +--rw max-threshold-unit?     identityref
          |                  |  +--rw weight?                 uint8
          |                  |  +--rw max-probability-val?    uint64
          |                  |  +--rw max-probability-unit?   identityref
          |                  +--rw ecn-enabled?   boolean
          +--:(scheduler-inline)
             +--rw scheduler
                +--rw min-rate
                |  +--rw value?   uint64
                |  +--rw unit?    identityref
                +--rw max-rate
                   +--rw value?         uint64
                   +--rw unit?          identityref
```

```
                  +--rw burst-value?   uint64
                  +--rw burst-unit?    identityref
```

Figure 17: DiffServ Module Tree Diagram

A.4. QoS Operational Module Tree Diagram

```
   module: ietf-qos-oper

     augment /if:interfaces/if:interface:
       +--ro qos-interface-statistics
          +--ro stats-per-direction* []
          |  +--ro direction?     identityref
          |  +--ro policy-name?   string
          |  +--ro classifier* []
          |  |  +--ro name?               string
          |  |  +--ro classified-pkts?    yang:zero-based-counter64
          |  |  +--ro classified-bytes?   yang:zero-based-counter64
          |  |  +--ro classified-rate?    yang:gauge64
          |  +--ro named* []
          |  |  +--ro name?             string
          |  |  +--ro aggregated
          |  |  |  +--ro pkts?    yang:zero-based-counter64
          |  |  |  +--ro bytes?   yang:zero-based-counter64
          |  |  |  +--ro rate?    yang:gauge64
          |  |  +--ro non-aggregated
          |  |     +--ro classifier* []
          |  |     |  +--ro name?               string
          |  |     |  +--ro classified-pkts?
          |  |     |  |       yang:zero-based-counter64
          |  |     |  +--ro classified-bytes?
          |  |     |  |       yang:zero-based-counter64
          |  |     |  +--ro classified-rate?    yang:gauge64
          |  |     +--ro metering* []
          |  |     |  +--ro meter-id?        string
          |  |     |  +--ro conform-pkts?    yang:zero-based-counter64
          |  |     |  +--ro conform-bytes?   yang:zero-based-counter64
          |  |     |  +--ro conform-rate?    yang:gauge64
          |  |     |  +--ro exceed-pkts?     yang:zero-based-counter64
          |  |     |  +--ro exceed-bytes?    yang:zero-based-counter64
          |  |     |  +--ro exceed-rate?     yang:gauge64
          |  |     |  +--ro violate-pkts?    yang:zero-based-counter64
          |  |     |  +--ro violate-bytes?   yang:zero-based-counter64
          |  |     |  +--ro violate-rate?    yang:gauge64
          |  |     |  +--ro drop-pkts?       yang:zero-based-counter64
          |  |     |  +--ro drop-bytes?      yang:zero-based-counter64
          |  |     +--ro queueing* []
          |  |        +--ro queue-id?                    string
```

```
|  |        +--ro transmit-pkts?
|  |        |       yang:zero-based-counter64
|  |        +--ro transmit-bytes?
|  |        |       yang:zero-based-counter64
|  |        +--ro queue-current-size-bytes?   yang:gauge64
|  |        +--ro queue-average-size-bytes?   yang:gauge64
|  |        +--ro queue-peak-size-bytes?      yang:gauge64
|  |        +--ro tail-drop-pkts?
|  |        |       yang:zero-based-counter64
|  |        +--ro tail-drop-bytes?
|  |        |       yang:zero-based-counter64
|  |        +--ro red-drop-pkts?
|  |        |       yang:zero-based-counter64
|  |        +--ro red-drop-bytes?
|  |        |       yang:zero-based-counter64
|  |        +--ro red-ecn-marked-pkts?
|  |        |       yang:zero-based-counter64
|  |        +--ro red-ecn-marked-bytes?
|  |        |       yang:zero-based-counter64
|  |        +--ro wred-stats* []
|  |           +--ro profile?            uint8
|  |           +--ro drop-pkts?
|  |           |       yang:zero-based-counter64
|  |           +--ro drop-bytes?
|  |           |       yang:zero-based-counter64
|  |           +--ro ecn-marked-pkts?
|  |           |       yang:zero-based-counter64
|  |           +--ro ecn-marked-bytes?
|  |                   yang:zero-based-counter64
|  +--ro metering* []
|  |  +--ro meter-id?        string
|  |  +--ro conform-pkts?    yang:zero-based-counter64
|  |  +--ro conform-bytes?   yang:zero-based-counter64
|  |  +--ro conform-rate?    yang:gauge64
|  |  +--ro exceed-pkts?     yang:zero-based-counter64
|  |  +--ro exceed-bytes?    yang:zero-based-counter64
|  |  +--ro exceed-rate?     yang:gauge64
|  |  +--ro violate-pkts?    yang:zero-based-counter64
|  |  +--ro violate-bytes?   yang:zero-based-counter64
|  |  +--ro violate-rate?    yang:gauge64
|  |  +--ro drop-pkts?       yang:zero-based-counter64
|  |  +--ro drop-bytes?      yang:zero-based-counter64
|  +--ro queueing* []
|     +--ro queue-id?                   string
|     +--ro transmit-pkts?
|     |       yang:zero-based-counter64
|     +--ro transmit-bytes?
|     |       yang:zero-based-counter64
```

```
          |      +--ro queue-current-size-bytes?   yang:gauge64
          |      +--ro queue-average-size-bytes?   yang:gauge64
          |      +--ro queue-peak-size-bytes?      yang:gauge64
          |      +--ro tail-drop-pkts?
          |      |       yang:zero-based-counter64
          |      +--ro tail-drop-bytes?
          |      |       yang:zero-based-counter64
          |      +--ro red-drop-pkts?
          |      |       yang:zero-based-counter64
          |      +--ro red-drop-bytes?
          |      |       yang:zero-based-counter64
          |      +--ro red-ecn-marked-pkts?
          |      |       yang:zero-based-counter64
          |      +--ro red-ecn-marked-bytes?
          |      |       yang:zero-based-counter64
          |      +--ro wred-stats* []
          |         +--ro profile?            uint8
          |         +--ro drop-pkts?          yang:zero-based-counter64
          |         +--ro drop-bytes?         yang:zero-based-counter64
          |         +--ro ecn-marked-pkts?    yang:zero-based-counter64
          |         +--ro ecn-marked-bytes?   yang:zero-based-counter64
        +---x clear
           +---w input
           |  +---w category?     identityref
           |  +---w started-at?   yang:date-and-time
           +--ro output
              +--ro finished-at?   yang:date-and-time
```

Figure 18: QoS Operational Module Tree Diagram

A.5. Scheduler Policy Module Tree Diagram

```
module: ietf-scheduler-policy

  augment /traffic-policy:policies/traffic-policy:policy
            /traffic-policy:classifier/traffic-policy:inline
            /traffic-policy:filter:
    +--rw (filter-params)?
       +--:(filter-match-all)
          +--rw match-all
             +--rw action?   empty
  augment /traffic-policy:policies/traffic-policy:policy
            /traffic-policy:classifier/traffic-policy:action:
    +--rw (action-params)?
       +--:(scheduler)
       |  +--rw scheduler
       |     +--rw min-rate
       |     |  +--rw value?   uint64
       |     |  +--rw unit?    identityref
       |     +--rw max-rate
       |        +--rw value?         uint64
       |        +--rw unit?          identityref
       |        +--rw burst-value?   uint64
       |        +--rw burst-unit?    identityref
       +--:(queue-policy-name)
          +--rw queue-policy-name
             +--rw queue-policy
                     -> /traffic-policy:policies/policy/name
```

Figure 19: Scheduler Policy Module Tree Diagram

A.6. Queue Policy Module Tree Diagram

```
module: ietf-queue-policy

  augment /traffic-policy:policies/traffic-policy:policy
            /traffic-policy:classifier/traffic-policy:inline
            /traffic-policy:filter:
    +--rw (filter-params)?
       +--:(traffic-group)
       |  +--rw traffic-group
       |     +--rw name?   string
       +--:(filter-match-all)
          +--rw match-all
             +--rw action?   empty
  augment /traffic-policy:policies/traffic-policy:policy
            /traffic-policy:classifier/traffic-policy:action:
    +--rw (action-params)?
       +--:(queue-template-name)
       |  +--rw queue-reference
```

```
          |     +--rw queue-name    string
          +--:(queue-inline)
             +--rw queue
                +--rw priority
                |  +--rw level?   uint8
                +--rw min-rate
                |  +--rw value?   uint64
                |  +--rw unit?    identityref
                +--rw max-rate
                |  +--rw value?         uint64
                |  +--rw unit?          identityref
                |  +--rw burst-value?   uint64
                |  +--rw burst-unit?    identityref
                +--rw algorithmic-drop
                   +--rw (drop-algorithm)?
                      +--:(tail-drop)
                      |  +--rw tail-drop
                      |     +--rw tail-drop?   empty
                      +--:(red)
                      |  +--rw red
                      |     +--rw min-threshold-val?      uint64
                      |     +--rw min-threshold-unit?     identityref
                      |     +--rw max-threshold-val?      uint64
                      |     +--rw max-threshold-unit?     identityref
                      |     +--rw weight?                 uint8
                      |     +--rw max-probability-val?    uint64
                      |     +--rw max-probability-unit?   identityref
                      |     +--rw ecn-enabled?            boolean
                      +--:(wred)
                         +--rw wred
                            +--rw wred* [profile]
                            |  +--rw profile                 uint8
                            |  +--rw color-type?             identityref
                            |  +--rw color-val* [min max]
                            |  |  +--rw min    uint8
                            |  |  +--rw max    uint8
                            |  +--rw min-threshold-val?      uint64
                            |  +--rw min-threshold-unit?     identityref
                            |  +--rw max-threshold-val?      uint64
                            |  +--rw max-threshold-unit?     identityref
                            |  +--rw weight?                 uint8
                            |  +--rw max-probability-val?    uint64
                            |  +--rw max-probability-unit?   identityref
                            +--rw ecn-enabled?   boolean
```

Figure 20: Queue Policy Module Tree Diagram

Appendix B. Company A and Company B examples

Company A and Company B DiffServ modules augments all the filter types of the QoS Classifier module as well as the QoS Policy module that allow it to define marking, metering, min-rate, max-rate actions. Queueing and metering counters are realized by augmenting of the QoS Target module.

B.1. Example of Company A DiffServ Model

The following Company A vendor example augments the QoS and DiffServ model, demonstrating some of the following functionality:

- use of template based classifier definitions

- use of single policy type modelling queue, scheduler policy, and a filter policy. All of these policies either augment the QoS policy or the DiffServ modules

- use of inline actions in a policy

- flexibility in marking dscp or metadata at ingress and/or egress.

```
module example-compa-diffserv {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:example-compa-diffserv";
  prefix example;

  import iana-qos-types {
    prefix qos-types;
    reference
      "RFC XXXX: YANG Models for Quality of Service (QoS).";
  }
  import ietf-traffic-policy {
    prefix traffic-policy;
    reference
      "RFC XXXX: YANG Models for Quality of Service (QoS).";
  }
  import ietf-qos-action {
    prefix qos-action;
    reference
      "RFC XXXX: YANG Models for Quality of Service (QoS).";
  }
  import ietf-diffserv {
    prefix diffserv;
    reference
      "RFC XXXX: YANG Models for Quality of Service (QoS).";
  }
```

```
     organization "Company A";
     contact
       "Editor:   XYZ
                 <mailto:xyz@compa.com>";
     description
       "This module contains a collection of YANG definitions of
        companyA diffserv specification extension.";
     revision 2024-01-31 {
       description
         "Initial revision for diffserv actions on network packets.";
       reference
         "RFC XXXX: A YANG Data Model for Quality of Service (QoS).";
     }

     identity default-policy-type {
       base qos-types:policy-type;
       description
         "This defines default policy-type.";
     }

     identity qos-group {
       base qos-types:filter-type;
       description
         "Filter-type to match on  packet metadata. This metadata
          can be used for packet marking, queuing for traffic in
          egress direction.";
     }

     grouping qos-group {
       list qos-group {
         key "min max";
         description
           "List of qos-groups to match on. A QoS group is a set of
            packets tagged as group.";
         leaf min {
           type uint8;
           description
             "Minimum value of qos-group range.";
         }
         leaf max {
           type uint8;
           must ". >= ../min" {
             error-message
             "The qos-group max must be
              greater than or equal to qos-group min.";
           }
           description
             "Maximum value of qos-group range.";
```

```
        }
      }
      description
        "Filter containing list of qos-group ranges.";
    }
    grouping bw-remaining-percent {
      container bw-remaining-percent {
        leaf value {
          type uint8;
          units "percentage";
          description
            "Percentage of the remaining bandwidth present.";
        }
        description
          "Remaining bandwidth of the link in percent.";
      }
      description
        "Grouping for bandwidth remaining after
        reserved and priority bandwidth.";
    }
    grouping qos-group-marking {
      container qos-group {
        leaf qos-group {
          type uint8;
          description
            "Mark metadata information in the network packet";
        }
        description
          "Container for QoS-Group marking.";
      }
      description
        "Grouping for QoS-Group marking.";
    }

    augment "/traffic-policy:classifiers/" +
      "traffic-policy:classifier/" +
      "traffic-policy:filter/diffserv:filter-param" {
      case qos-group {
        uses qos-group;
        description
          "Filter containing list of qos-group ranges.
          QoS-group represent packet metadata information
          in a network packet.";
      }
      description
        "Augmentation of classifier filters.";
    }
```

```
     augment "/traffic-policy:policies/traffic-policy:policy/" +
       "traffic-policy:classifier/" +
       "traffic-policy:action" {
       choice action-params {
         case priority {
           uses qos-action:priority;
         }
         case min-rate {
           uses qos-action:min-rate;
         }
         case max-rate {
           uses qos-action:max-rate;
         }
         case bw-remaining {
           uses bw-remaining-percent;
         }
         case mark-qos-group {
           uses qos-group-marking;
         }
         description
           "Choice of action parameters.";
       }
       description
         "Augments policy entry to add action parameters.";
     }

     augment "/traffic-policy:policies" +
       "/traffic-policy:policy" +
       "/traffic-policy:classifier" +
       "/traffic-policy:action" +
       "/diffserv:action-params" +
       "/diffserv:meter-inline" +
       "/diffserv:meter-types" +
       "/diffserv:single-rate-two-color-meter-type" +
       "/diffserv:single-rate-two-color-meter" {
       description
         "Augment the single-rate-two-color meter to add" +
         "color classifiers.";
       container conform-color {
         uses traffic-policy:generic-classifier-attr;
         description
           "Conform color classifier container.";
       }
       container exceed-color {
         uses traffic-policy:generic-classifier-attr;
         description
           "Exceed color classifier container.";
       }
```

```
     }

     augment "/traffic-policy:policies" +
       "/traffic-policy:policy" +
       "/traffic-policy:classifier" +
       "/traffic-policy:action" +
       "/diffserv:action-params" +
       "/diffserv:meter-inline" +
       "/diffserv:meter-types" +
       "/diffserv:single-rate-three-color-meter-type" +
       "/diffserv:single-rate-three-color-meter" {
       description
         "Augment the one-rate-tri-color meter to add" +
         "color classifiers.";
       container conform-color {
         uses traffic-policy:generic-classifier-attr;
         description
           "Conform color classifier container.";
       }
       container exceed-color {
         uses traffic-policy:generic-classifier-attr;
         description
           "Exceed color classifier container.";
       }
       container violate-color {
         uses traffic-policy:generic-classifier-attr;
         description
           "Violate color classifier container.";
       }
     }

     augment "/traffic-policy:policies" +
       "/traffic-policy:policy" +
       "/traffic-policy:classifier" +
       "/traffic-policy:action" +
       "/diffserv:action-params" +
       "/diffserv:meter-inline" +
       "/diffserv:meter-types" +
       "/diffserv:two-rate-three-color-meter-type" +
       "/diffserv:two-rate-three-color-meter" {
       description
         "Augment the two-rate-tri-color meter to add" +
         "color classifiers.";
       container conform-color {
         uses traffic-policy:generic-classifier-attr;
         description
           "Conform color classifier container.";
       }
```

```
      container exceed-color {
        uses traffic-policy:generic-classifier-attr;
        description
          "Exceed color classifier container.";
      }
      container violate-color {
        uses traffic-policy:generic-classifier-attr;
        description
          "Violate color classifier container.";
      }
    }
  }
```

B.2. Example of Company B DiffServ Model

The following vendor example augments the QoS and DiffServ model, demonstrating some of the following functionality:

- use of inline classifier definitions (defined inline in the policy vs referencing an externally defined classifier)

- use of mulitple policy types, e.g. a queue policy, a scheduler policy, and a filter policy. All of these policies either augment the QoS policy or the DiffServ modules

- use of a queue module, which uses and extends the queue grouping from the QoS Action module

- use of meter templates (v.s. meter inline)

- use of internal meta data for classification and marking

```
module example-compb-diffserv-filter-policy {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:" +
            "example-compb-diffserv-filter-policy";
  prefix compb-filter-policy;

  import iana-qos-types {
    prefix qos-types;
    reference
      "RFC XXXX: YANG Models for Quality of Service (QoS).";
  }
  import ietf-traffic-policy {
    prefix traffic-policy;
  }
  import ietf-qos-action {
    prefix qos-action;
```

```
     }
     import ietf-diffserv {
       prefix diffserv;
     }

     organization
       "Company B";

     contact
       "Editor:   XYZ
                  <mailto:xyz@compb.com>";

     description
       "This module contains a collection of YANG definitions for
        configuring diffserv specification implementations.

        Copyright (c) 2022 IETF Trust and the persons identified as
        authors of the code.  All rights reserved.

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

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

     revision 2024-01-31 {
       description
         "Initial revision of diffserv policy for Company B.";
       reference
         "RFC XXXX: YANG Data Model for QoS.";
     }


     /*************************************************
      * Classification types
      *************************************************/

     identity internal-loss-priority  {
        base qos-types:filter-type;
        description
           "Internal loss priority filter type.";
     }

     grouping loss-priority {
```

```
      list loss-priority {
        key "loss-priority";
        description
          "list of loss-priorities";
        leaf loss-priority {
          type enumeration {
            enum high {
             description "High loss Priority.";
            }
            enum medium-high {
             description "Medium to high loss priority.";
            }
            enum medium-low {
             description "Medium to low loss priority.";
            }
            enum low {
             description "Low loss priority.";
            }
          }
          description
            "Loss priority.";
        }
      }
      description
        "Filter containing list of loss priorities";
    }

    augment "/traffic-policy:policies" +
            "/traffic-policy:policy" +
            "/traffic-policy:classifier" +
            "/traffic-policy:inline/traffic-policy:filter" +
            "/diffserv:filter-params" {
      case internal-loss-priority {
          uses loss-priority;
        description
          "Filter Type Internal-loss-priority";
      }
      description
        "Augments Diffserv Classifier with vendor " +
        "specific types.";
    }

    /***************************************************
     * Actions
     ***************************************************/

    identity mark-loss-priority {
      base qos-types:action-type;
```

```
      description
        "Mark loss-priority action type.";
    }

    grouping mark-loss-priority {
      container mark-loss-priority {
        leaf loss-priority {
          type enumeration {
            enum high {
             description "High loss Priority.";
            }
            enum medium-high {
             description "Medium to high loss priority.";
            }
            enum medium-low {
             description "Medium to low loss priority.";
            }
            enum low {
             description "Low loss priority.";
            }
          }
          description
            "Loss priority.";
        }
        description
          "Mark loss priority container.";
      }
      description
        "Mark loss priority grouping.";
    }

    augment "/traffic-policy:policies" +
            "/traffic-policy:policy" +
            "/traffic-policy:classifier" +
            "/traffic-policy:action" +
            "/diffserv:action-params" {
      case traffic-group-marking {
        uses qos-action:traffic-group-marking;
        description
          "Mark traffic group in the packet.";
      }
      case mark-loss-priority {
          uses mark-loss-priority;
        description
          "Mark loss priority in the packet.";
      }
      case meter-reference {
          uses qos-action:meter-reference;
```

```
        description
          "Assign a meter as an action.";
      }
      case discard {
          uses qos-action:discard;
        description
          "Discard action.";
      }
      case count {
          uses qos-action:count;
        description
          "Count action - explicit count configuration.";
      }

      description
        "Augments common diffserv policy actions";
    }

    augment "/qos-action:meters" +
            "/qos-action:meter" +
            "/qos-action:meter-types" +
            "/qos-action:single-rate-three-color-meter-type" +
            "/qos-action:single-rate-three-color-meter" {
       leaf one-rate-color-aware {
        type boolean;
        description
          "This defines if the meter is color-aware.";
      }
      description
        "Augmentation of color-aware flag.";
    }
    augment "/qos-action:meters" +
            "/qos-action:meter" +
            "/qos-action:meter-types" +
            "/qos-action:two-rate-three-color-meter-type" +
            "/qos-action:two-rate-three-color-meter" {
       leaf two-rate-color-aware {
        type boolean;
        description
          "This defines if the meter is color-aware.";
      }
      description
        "Augmentation of color-aware flag.";
    }

  }

  module example-compb-queue-policy {
```

```
yang-version 1.1;
namespace "urn:ietf:params:xml:ns:yang:example-compb-queue-policy";
prefix queue-plcy;

import iana-qos-types {
  prefix qos-types;
  reference
    "RFC XXXX: YANG Models for Quality of Service (QoS).";
}
import ietf-traffic-policy {
  prefix traffic-policy;
}

organization "Company B";
contact
  "Editor:   XYZ
            <mailto:xyz@compb.com>";

description
  "This module defines a queue policy. The classification
   is based on aforwarding class, and the actions are queues.

   Copyright (c) 2014 IETF Trust and the persons identified as
   authors of the code.  All rights reserved.

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

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

revision 2024-01-31 {
  description
    "Initial revision of diffserv policy";
  reference
    "RFC XXXX: A YANG Data Model for Quality of Service (QoS).";
}

identity compb-queue {
  base qos-types:action-type;
  description
    "compb-queue action type";
}
```

```
     grouping compb-queue-name {
       container queue-name {
         leaf name {
           type string;
           description
             "Queue class name";
         }
         description
           "compb queue container";
       }
       description
         "compb-queue grouping";
     }

     augment "/traffic-policy:policies" +
             "/traffic-policy:policy" +
             "/traffic-policy:classifier" +
             "/traffic-policy:action" {
       choice action-params {
         description
           "Choice of action types";
         case compb-queue {
           uses compb-queue-name;
         }
       }
       description
         "Augment the queue actions to queue policy entry";
     }
   }

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

     import ietf-qos-action {
       prefix qos-action;
       reference "RFC XXXX: YANG Model for QoS";
     }

     organization "Company B";
     contact
       "Editor:   XYZ
                  <mailto:xyz@compb.com>";

     description
       "This module describes a compb queue module. This is a
        template for a queue within a queue policy, referenced
```

```
        by name.

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

     revision 2024-01-31 {
       description
         "Initial revision of diffserv based classifier";
       reference
         "RFC XXXX: YANG Data Model for Quality of Service (QoS).";
     }

     container compb-queue {
       description
         "Queue used in compb architecture";
       leaf name {
         type string;
         description
           "A unique name identifying this queue";
       }
       uses qos-action:queue;
       container excess-rate {
         choice excess-rate-type {
           case percent {
             leaf excess-rate-percent {
               type uint32 {
                 range "1..100";
               }
               description
                 "excess-rate-percent";
             }
           }
           case proportion {
             leaf excess-rate-poroportion {
               type uint32 {
                 range "1..1000";
               }
               description
                 "excess-rate-poroportion";
             }
           }
           description
             "Choice of excess-rate type";
         }
         description
           "Excess rate value";
       }
       leaf excess-priority {
```

```
        type enumeration {
          enum high {
            description "High Loss Priority";
          }
          enum medium-high {
            description "Medium-high Loss Priority";
          }
          enum medium-low {
            description "Medium-low Loss Priority";
          }
          enum low {
            description "Low Loss Priority";
          }
          enum none {
            description "No excess priority";
          }
        }
        description
          "Priority of excess (above guaranted rate) traffic";
      }
      container buffer-size {
        choice buffer-size-type {
          case percent {
            leaf buffer-size-percent {
              type uint32 {
                range "1..100";
              }
              description
                "buffer-size-percent";
            }
          }
          case temporal {
            leaf buffer-size-temporal {
              type uint64;
              units "microsecond";
              description
                "buffer-size-temporal";
            }
          }
          case remainder {
            leaf buffer-size-remainder {
              type empty;
              description
                "use remaining of buffer";
            }
          }
          description
            "Choice of buffer size type";
```

```
        }
        description
          "Buffer size value";
      }
    }

    augment
      "/compb-queue/queue/algorithmic-drop" +
      "/drop-algorithm" {
      case random-detect {
        list drop-profile-list {
          key "priority";
          description
            "map of priorities to drop-algorithms";
          leaf priority {
            type enumeration {
              enum any {
                description "Any priority mapped here";
              }
              enum high {
                description "High Priority Packet";
              }
              enum medium-high {
                description "Medium-high Priority Packet";
              }
              enum medium-low {
                description "Medium-low Priority Packet";
              }
              enum low {
                description "Low Priority Packet";
              }
            }
            description
              "Priority of guaranteed traffic";
          }
          leaf drop-profile {
            type string;
            description
              "drop profile to use for this priority";
          }
        }
      }
      description
        "compb random detect drop algorithm config";
    }
  }

  module example-compb-scheduler-policy {
```

```
yang-version 1.1;
namespace "urn:ietf:params:xml:ns:yang:" +
          "example-compb-scheduler-policy";
prefix scheduler-plcy;

import ietf-qos-action {
  prefix qos-action;
  reference
    "RFC XXXX: A YANG Data Model for Quality of Service (QoS).";
}
import ietf-traffic-policy {
  prefix traffic-policy;
  reference
    "RFC XXXX: A YANG Data Model for Quality of Service (QoS).";
}

organization
  "Company B";

contact
  "Editor:   XYZ
             <mailto:xyz@compb.com>";

description
  "This module defines a scheduler policy. The classification
   is based on classifier-any, and the action is a scheduler.";

revision 2024-01-31 {
  description
    "Initial revision of diffserv policy";
  reference
    "RFC XXXX: A YANG Data Model for Quality of Service (QoS).";
}

grouping queue-policy-name {
  container compb-queue-policy-name {
    leaf name {
      type string;
      description
        "Queue policy name";
    }
    description
      "compb-queue-policy container";
  }
  description
    "compb-queue policy grouping";
}
```

```
     augment "/traffic-policy:policies" +
       "/traffic-policy:policy" +
       "/traffic-policy:classifier" +
       "/traffic-policy:action" {
       choice action-params {
         case scheduler {
           uses qos-action:scheduler;
         }
         case queue-policy {
           uses queue-policy-name;
         }
         description
           "Augment the scheduler policy with a queue policy";
       }
       description
         "Augmentation of the scheduler policy with a queue policy";
     }
   }
```

Appendix C. Configuration examples

This section describes several examples of how the models defined in this draft can be used for configuration.

C.1. Configuration example for QoS Classifier

```
<!--
    This example shows a QoS classifier configuration.
    This classifier will match on any packet which has
    dscp value not in the range of 11-13 and source-port
    value which is in the range of either 10000-10300 or
    17540-19800. In other words, packets with dscp value
    0-10 or 14-63 and source-port value 10000-10300 or
    17540-19800
-->
<?xml version="1.0" encoding="UTF-8"?>
<classifiers
    xmlns="urn:ietf:params:xml:ns:yang:ietf-traffic-policy"
    xmlns:qt="urn:ietf:params:xml:ns:yang:iana-qos-types">
  <classifier>
    <name>my-classifier</name>
    <filter-operation>match-all</filter-operation>
    <filter>
      <type>qt:dscp</type>
      <logical-not>true</logical-not>
      <dscp
        xmlns="urn:ietf:params:xml:ns:yang:ietf-diffserv">
        <min>11</min>
        <max>13</max>
      </dscp>
    </filter>
    <filter>
      <type>qt:source-port</type>
      <logical-not>false</logical-not>
        <source-port
          xmlns="urn:ietf:params:xml:ns:yang:ietf-diffserv">
          <min>10000</min>
          <max>10300</max>
        </source-port>
        <source-port
          xmlns="urn:ietf:params:xml:ns:yang:ietf-diffserv">
          <min>17540</min>
          <max>19800</max>
        </source-port>
    </filter>
  </classifier>
</classifiers>
```

Figure 21: Configuration example for QoS Classifier

C.2. Configuration example for QoS Policy

```
<!--
    This example shows a QoS policy configuration.
-->

<?xml version="1.0" encoding="UTF-8"?>
<policies
     xmlns="urn:ietf:params:xml:ns:yang:ietf-traffic-policy"
     xmlns:qt="urn:ietf:params:xml:ns:yang:iana-qos-types">
  <policy>
    <name>my-policy</name>
    <type>qt:diffserv-policy-type</type>
    <classifier>
      <name>my-classifier</name>
      <inline>
        <filter>
              <type>qt:dscp</type>
              <logical-not>false</logical-not>
            <dscp
              xmlns="urn:ietf:params:xml:ns:yang:ietf-diffserv">
              <min>21</min>
              <max>22</max>
            </dscp>
        </filter>
      </inline>
      <action>
        <type>qt:dscp-marking</type>
        <dscp
          xmlns="urn:ietf:params:xml:ns:yang:ietf-diffserv">
          <dscp>23</dscp>
        </dscp>
      </action>
    </classifier>
  </policy>
</policies>
```

Figure 22: Configuration example for QoS Policy

C.3. Configuration example for QoS Policing

```
<!--
    This example shows a QoS policy configuration for a single rate
    two color policing. For a description of the problem refer to:
    https://networklessons.com/quality-of-service/
            policing-configuration-example
-->

<?xml version="1.0" encoding="UTF-8"?>
<classifiers
```

```
    xmlns="urn:ietf:params:xml:ns:yang:ietf-traffic-policy"
    xmlns:qt="urn:ietf:params:xml:ns:yang:iana-qos-types">
  <classifier>
    <name>ICMP</name>
    <description>class-map definition for ICMP</description>
    <filter>
      <type>qt:protocol</type>
      <logical-not>false</logical-not>
      <protocol
          xmlns="urn:ietf:params:xml:ns:yang:ietf-diffserv">
        <min>4</min>
        <max>4</max>
      </protocol>
    </filter>
  </classifier>
</classifiers>

<policies
    xmlns="urn:ietf:params:xml:ns:yang:ietf-traffic-policy"
    xmlns:qt="urn:ietf:params:xml:ns:yang:iana-qos-types">
  <policy>
    <name>SINGLE-RATE-TWO-COLOR</name>
    <type>qt:diffserv-policy-type</type>
    <classifier>
      <name>ICMP</name>
      <action>
        <type>qt:meter-inline</type>
        <single-rate-two-color-meter
            xmlns="urn:ietf:params:xml:ns:yang:ietf-diffserv">
          <committed-information-rate>128000</committed-information-rate>
          <conform-action>
            <action-type>qt:action-transmit</action-type>
          </conform-action>
          <exceed-action>
            <action-type>qt:action-drop</action-type>
          </exceed-action>
        </single-rate-two-color-meter>
      </action>
    </classifier>
  </policy>
</policies>

<interfaces
    xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces"
    xmlns:qt="urn:ietf:params:xml:ns:yang:iana-qos-types"
    xmlns:ianaift="urn:ietf:params:xml:ns:yang:iana-if-type">
  <interface>
    <name>FastEthernet 0/0</name>
```

```
    <type>ianaift:ethernetCsmacd</type>
    <qos-target-policy
        xmlns="urn:ietf:params:xml:ns:yang:ietf-traffic-policy">
      <direction>qt:ingress</direction>
      <type>qt:diffserv-policy-type</type>
      <name>SINGLE-RATE-TWO-COLOR</name>
    </qos-target-policy>
  </interface>
</interfaces>
```

Figure 23: Configuration example for QoS Policing

C.4. Configuration example for QoS Queueing

```
   <!--
       This example shows a QoS policy configuration for queuing.
       For a description of the problem refer to:
       https://networklessons.com/quality-of-service/
               queuing-configuration-example
   -->

   <?xml version="1.0" encoding="UTF-8"?>
   <classifiers
       xmlns="urn:ietf:params:xml:ns:yang:ietf-traffic-policy"
       xmlns:qt="urn:ietf:params:xml:ns:yang:iana-qos-types">
     <classifier>
       <name>VOICE</name>
       <description>class-map definition for VOICE</description>
       <filter>
         <type>qt:protocol</type>
         <logical-not>false</logical-not>
         <dscp
             xmlns="urn:ietf:params:xml:ns:yang:ietf-diffserv">
           <min>46</min>
           <max>46</max>
         </dscp>
       </filter>
     </classifier>
     <classifier>
       <name>VIDEO</name>
       <description>class-map definition for VIDEO</description>
       <filter>
         <type>qt:protocol</type>
         <logical-not>false</logical-not>
         <dscp
             xmlns="urn:ietf:params:xml:ns:yang:ietf-diffserv">
           <min>34</min>
           <max>34</max>
```

```
        </dscp>
      </filter>
    </classifier>
    <classifier>
      <name>DATA</name>
      <description>class-map definition for DATA</description>
      <filter>
        <type>qt:protocol</type>
        <logical-not>false</logical-not>
        <dscp
            xmlns="urn:ietf:params:xml:ns:yang:ietf-diffserv">
          <min>10</min>
          <max>10</max>
        </dscp>
      </filter>
    </classifier>
  </classifiers>

  <policies
      xmlns="urn:ietf:params:xml:ns:yang:ietf-traffic-policy"
      xmlns:qt="urn:ietf:params:xml:ns:yang:iana-qos-types">
    <policy>
      <name>QUEUEING-POLICY</name>
      <type>qt:diffserv-policy-type</type>
      <classifier>
        <name>VOICE</name>
        <action>
          <type>qt:queue</type>
          <queue
              xmlns="urn:ietf:params:xml:ns:yang:ietf-diffserv">
            <priority>
              <level>1</level>
            </priority>
            <min-rate>
              <value>2000000</value>
              <unit>qt:bits-per-second</unit>
            </min-rate>
            <max-rate>
              <value>4000000</value>
              <unit>qt:bits-per-second</unit>
            </max-rate>
          </queue>
        </action>
      </classifier>
      <classifier>
        <name>VIDEO</name>
        <action>
          <type>qt:queue</type>
```

```
          <queue
              xmlns="urn:ietf:params:xml:ns:yang:ietf-diffserv">
            <priority>
              <level>2</level>
            </priority>
            <min-rate>
              <value>10000000</value>
              <unit>qt:bits-per-second</unit>
            </min-rate>
            <max-rate>
              <value>30000000</value>
              <unit>qt:bits-per-second</unit>
            </max-rate>
          </queue>
        </action>
      </classifier>
      <classifier>
        <name>DATA</name>
        <action>
          <type>qt:queue</type>
          <queue
              xmlns="urn:ietf:params:xml:ns:yang:ietf-diffserv">
            <priority>
              <level>2</level>
            </priority>
            <min-rate>
              <value>8000000</value>
              <unit>qt:bits-per-second</unit>
            </min-rate>
            <max-rate>
              <value>18000000</value>
              <unit>qt:bits-per-second</unit>
            </max-rate>
          </queue>
        </action>
      </classifier>
    </policy>
  </policies>

  <interfaces
      xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces"
      xmlns:qt="urn:ietf:params:xml:ns:yang:iana-qos-types"
      xmlns:ianaift="urn:ietf:params:xml:ns:yang:iana-if-type">
    <interface>
      <name>FastEthernet 0/0</name>
      <type>ianaift:ethernetCsmacd</type>
      <qos-target-policy
          xmlns="urn:ietf:params:xml:ns:yang:ietf-traffic-policy">
```

```
        <direction>qt:egress</direction>
        <type>qt:diffserv-policy-type</type>
        <name>QUEUEING-POLICY</name>
      </qos-target-policy>
    </interface>
  </interfaces>
```

Figure 24: Configuration example for QoS Queueing

C.5. Configuration example for QoS Scheduling

```
<!--
    This example shows a QoS policy configuration for scheduling.
-->

<?xml version="1.0" encoding="UTF-8"?>

<policies
    xmlns="urn:ietf:params:xml:ns:yang:ietf-traffic-policy"
    xmlns:qp="urn:ietf:params:xml:ns:yang:ietf-queue-policy"
    xmlns:sp="urn:ietf:params:xml:ns:yang:ietf-scheduler-policy"
    xmlns:ds="urn:ietf:params:xml:ns:yang:ietf-diffserv"
    xmlns:qt="urn:ietf:params:xml:ns:yang:iana-qos-types">
  <policy>
    <name>QUEUEING-POLICY</name>
    <type>qt:queue-policy-type</type>
    <classifier>
      <name>VOICE</name>
      <inline>
        <filter>
          <type>qt:traffic-group-name</type>
          <logical-not>false</logical-not>
          <traffic-group
              xmlns="urn:ietf:params:xml:ns:yang:ietf-queue-policy">
            <name>voice</name>
          </traffic-group>
        </filter>
      </inline>
      <action>
        <type>qt:queue</type>
        <queue
            xmlns="urn:ietf:params:xml:ns:yang:ietf-queue-policy">
          <priority>
            <level>1</level>
          </priority>
          <min-rate>
            <value>2000000</value>
            <unit>qt:bits-per-second</unit>
```

```
            </min-rate>
            <max-rate>
              <value>4000000</value>
              <unit>qt:bits-per-second</unit>
            </max-rate>
          </queue>
        </action>
      </classifier>
      <classifier>
        <name>VIDEO</name>
        <inline>
          <filter>
            <type>qt:traffic-group-name</type>
            <logical-not>false</logical-not>
            <traffic-group
                xmlns="urn:ietf:params:xml:ns:yang:ietf-queue-policy">
              <name>video</name>
            </traffic-group>
          </filter>
        </inline>
        <action>
          <type>qt:queue</type>
          <queue
              xmlns="urn:ietf:params:xml:ns:yang:ietf-queue-policy">
            <priority>
              <level>2</level>
            </priority>
            <min-rate>
              <value>10000000</value>
              <unit>qt:bits-per-second</unit>
            </min-rate>
            <max-rate>
              <value>30000000</value>
              <unit>qt:bits-per-second</unit>
            </max-rate>
          </queue>
        </action>
      </classifier>
      <classifier>
        <name>DATA</name>
        <inline>
          <filter>
            <type>qt:traffic-group-name</type>
            <logical-not>false</logical-not>
            <traffic-group
                xmlns="urn:ietf:params:xml:ns:yang:ietf-queue-policy">
              <name>data</name>
            </traffic-group>
```

```
          </filter>
        </inline>
        <action>
          <type>qt:queue</type>
          <queue
              xmlns="urn:ietf:params:xml:ns:yang:ietf-queue-policy">
            <priority>
              <level>2</level>
            </priority>
            <min-rate>
              <value>8000000</value>
              <unit>qt:bits-per-second</unit>
            </min-rate>
            <max-rate>
              <value>18000000</value>
              <unit>qt:bits-per-second</unit>
            </max-rate>
          </queue>
        </action>
      </classifier>
    </policy>
    <policy>
      <name>SCHEDULING-POLICY</name>
      <type>qt:scheduler-policy-type</type>
      <classifier>
        <name>ALL</name>
        <inline>
          <filter>
            <type>qt:filter-match-all</type>
            <logical-not>false</logical-not>
            <sp:match-all>
              <sp:action></sp:action>
            </sp:match-all>
          </filter>
        </inline>
        <action>
          <type>qt:scheduler</type>
          <scheduler
              xmlns="urn:ietf:params:xml:ns:yang:ietf-scheduler-policy">
            <min-rate>
              <value>20000000</value>
              <unit>qt:bits-per-second</unit>
            </min-rate>
            <max-rate>
              <value>50000000</value>
              <unit>qt:bits-per-second</unit>
            </max-rate>
          </scheduler>
```

```
        </action>
        <action>
          <type>qt:queue-policy-name</type>
          <queue-policy-name
              xmlns="urn:ietf:params:xml:ns:yang:ietf-scheduler-policy">
            <queue-policy>QUEUEING-POLICY</queue-policy>
          </queue-policy-name>
        </action>
      </classifier>
    </policy>
  </policies>

  <interfaces
      xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces"
      xmlns:qt="urn:ietf:params:xml:ns:yang:iana-qos-types"
      xmlns:ianaift="urn:ietf:params:xml:ns:yang:iana-if-type">
    <interface>
      <name>FastEthernet 0/0</name>
      <type>ianaift:ethernetCsmacd</type>
      <qos-target-policy
          xmlns="urn:ietf:params:xml:ns:yang:ietf-traffic-policy">
        <direction>qt:egress</direction>
        <type>qt:scheduler-policy-type</type>
        <name>SCHEDULING-POLICY</name>
      </qos-target-policy>
    </interface>
  </interfaces>
```

Figure 25: Configuration example for QoS Scheduling

Authors' Addresses

Aseem Choudhary
Aviatrix Systems
2901 Tasman Drive #109
Santa Clara, CA 95054
United States of America
Email: achoudhary@aviatrix.com

Mahesh Jethanandani
Kloud Services
Email: mjethanandani@gmail.com

Ebben Aries
Juniper Networks
1194 North Mathilda Avenue
Sunnyvale, CA 94089
United States of America
Email: exa@juniper.net

Ing-Wher Chen
The MITRE Corporation
Email: ingwherchen@mitre.org

RTGWG
Internet-Draft
Intended status: Informational
Expires: 27 October 2022

D. King
Lancaster University
A. Farrel
Old Dog Consulting
C. Jacquenet
Orange

25 April 2022

Challenges for the Internet Routing Systems Introduced by Semantic Routing

draft-king-irtf-challenges-in-routing-08

Abstract

Historically, the meaning of an IP address has been to identify an interface on a network device. Routing protocols were developed based on the assumption that a destination address had this semantic.

Over time, routing decisions have been enhanced to determine paths on which packets could be forwarded according to additional information carried principally within the packet headers, and dependent on policy coded in, configured at, or signaled to the routers.

Many proposals have been made to add semantics to IP packets by placing additional information into existing fields, by adding semantics to IP addresses, or by adding fields to the packets. The intent is always to facilitate routing decisions based on these additional semantics in order to provide differentiated paths to enable forwarding of different packet flows on paths that may be distinct from those derived by shortest path first or path vector routing. We call this approach "Semantic Routing".

This document describes the challenges to the existing routing system that are introduced by Semantic Routing. It then summarizes the opportunities for research into new or modified routing and forwarding approaches that make use of additional semantics.

This document is presented as a study to support further research into clarifying and understanding the issues. It does not pass comment on the advisability or practicality of any of the proposals and does not define any technical solutions.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

Historically, the meaning of an IP address has been to identify an interface on a network device. Routing protocols were to compute, establish, and maintain paths through networks toward destination prefixes until IP packets eventually reach their destination, and were based on the assumption that a destination address had this semantic. Anycast and multicast addresses were also defined, and those address semantics sometimes required variations to the routing protocols or even encouraged the development of new protocols.

Over time, the mechanisms that enabled routing decisions were enhanced to determine paths on which packets could be forwarded according to additional information carried principally within the packets headers or within 'shim' headers, and dependent on policy coded in, configured at, or signaled to the routers. Perhaps one of the most iconic examples is Equal-Cost Multipath (ECMP) where a router makes a choice about how to forward a packet over a number of parallel links or paths based on the values of a set of fields in the packet header.

Many proposals have been made to add semantics to IP packets by placing additional information into existing fields, by adding semantics to IP addresses, or by adding fields to the packets. The intent is always to facilitate routing decisions based on these additional semantics in order to provide differentiated paths to enable forwarding of different packet flows on paths that may be distinct from those derived by shortest path first or path vector routing. We call this approach "Semantic Routing" [I-D.farrel-irtf-introduction-to-semantic-routing].

There are many approaches to adding semantics to packet headers: the additional information may be derived from the destination addresses, from other fields in the packet header, or the packet itself. Mechanisms for using the destination address range from assigning an address prefix to have a special purpose and meaning (such as is done for multicast addressing) through allowing the owner of a prefix to use the low-order bits of an address for specific purposes (e.g., to provide an indication of the nature of the service that is associated with these packets). Some proposals suggest variable address lengths, others offer new hierarchical address formats, and some introduce a structure to addresses so that they can carry additional information in a common way. Alternatively, forwarding decisions can be performed based on fields in the packet header (such as the IPv6 Flow Label, or the Traffic Class field), overloading of existing packet fields, or new fields added to the packet headers.

A survey of ways in which routing and forwarding decisions have been made based on additional information carried in packets can be found in [I-D.king-irtf-semantic-routing-survey].

Some Semantic Routing proposals are intended to be deployed in administratively scoped IP domains whose network components (routers, switches, etc.) are operated by a single administrative entity (sometimes referred to as 'limited domains' [RFC8799]), while other proposals are intended for use across the Internet. The impact the proposals have on routing systems may require clean-slate solutions, hybrid solutions, extensions to existing routing protocols, or potentially no changes at all.

This document describes some of the key challenges to the routing system that are already present in today's IP networks. It then briefly outlines the concept of "Semantic Routing" with reference to [I-D.farrel-irtf-introduction-to-semantic-routing] and presents some of the additional challenges to the existing routing system that Semantic Routing may introduce. Finally, this document presents a list of research questions that offer opportunities for future research into new or modified routing protocols and forwarding systems that make use of Semantic Routing.

In this document, the focus is on routing and forwarding at the IP layer. A variety of overlay mechanisms exists to perform service or path routing at higher layers, and those approaches may be based on similar extensions to packet semantics, but that is out of scope for this document. Similarly, it is possible that Semantic Routing can be applied in a number of underlay network technologies, and that, too, is out of scope for this document.

This document is presented as a study to support further research into clarifying and understanding the issues. It does not pass comment on the advisability or practicality of any of the proposals and does not define any technical solutions.

2. Current Challenges to IP Routing

Today's IP routing faces several significant challenges which are a consequence of architectural design decisions and the continued exponential growth in traffic. These challenges include mobility, multihoming, programmable paths, scalability, and security, and were not the focus of the original design of the Internet. Nevertheless, IP networks have, in general, coped well in an incremental manner whenever a new challenge has arisen. The following list is presented to give context to the continuing requirements that routing protocols must meet as new semantics are applied to the routing process.

* Mobility - Mobility introduces several challenges, including maintaining a relationship between a sender and a receiver in cases where the sender or receiver changes their point of network attachment. The network must always be informed about the mobile node's current location, to allow continuity of services. Mobile users may also consume network resources, while in motion. The mobile user's service instances and attachments will also change due to varying load or latency, e.g., in Multi-access Edge Computing (MEC) environments.

* Multihoming - Multihomed stations or multihomed networks are connected to the Internet via more than one access circuit or access network and, therefore, may be assigned multiple IP addresses or prefixes from different pools. There are challenges concerning how traffic is forwarded back to the source if the source has originated its traffic using the wrong source address for a particular connection, or if one of the connections to the Internet is degraded.

* Multi-path - The Internet was initially designed to find the single, "best" path to a destination using a distributed routing algorithm. Current IP network topologies can provide multiple paths to reach a destination, each with different characteristics and with different failure likelihoods. It may be beneficial to send traffic over multiple paths to achieve reliability and enhance throughput, and it may be desirable to select one path or another because of QoS or security considerations for example, or to avoid transiting specific areas of an IP network, based (for example) on the reputation of transit provider for example. However, how packets are forwarded by using the shortest path means that distinguishing these alternate paths and directing traffic to them can be hard. Further, problems concerning scalability, commercial agreements among Service Providers, and the design of BGP make the utilization of multi-path techniques difficult for inter-domain routing. (Note that this discussion is distinct from Equal Cost Multi-path (ECMP) where packets are directed onto several "parallel" paths of identical least cost using a hash algorithm operated on some of the packets' header fields.)

* Multicast - Delivering the same packet to multiple destinations can place considerable load on a network. Solutions that replicate the packet at the source or at the network edge may obviously cause multiple copies of the packet to flow along the same network links. Solutions that move deterministic replication into the network to make more optimal use of the network resources can be complex to set up and manage since multicast network designs often assume dynamic tree computation where the multicast

distribution tree can be rooted at the source or in the multicast network, thereby leading to specific routing tables whose entries denote the tree structure. More complicated hardware that can replicate packets may also be required within the network. In order that packets can be addressed to a group of destinations and not be forwarded by means of unicast transmission, parts of the addressing space (that is, address prefixes) have been reserved for multicast addressing.

* Programmable Paths - The ability to decouple IP paths from routing protocols and agreements between Service Providers could allow users and applications to select network paths themselves, based on the required path characteristics. Another option is to let the route computation logic select, establish, and maintain paths on behalf of the user or the application and as a function of their requirements so that Service Providers can participate in the route computation "service". Currently, user and application packets follow the path selected by routing protocols and the way traffic is forwarded through a network is under the control of the Service Provider that operates the said network. The corresponding traffic forwarding policies enforced by the service provider usually comply with the requirements expressed by the user or the application. These requirements may have triggered a dynamic service parameter negotiation cycle that eventually leads to proper (network, CPU, storage) resource allocation.

* Endpoint Selection - As compute resources and content storage move closer to the edge of the network, there are often multiple points in the network that can satisfy user requests. In order to make the best use of these distributed resources and so as to not overload parts of the network, user traffic needs to be steered to appropriate servers or data centres. In many cases, this function may be achieved in the application layer (such as through DNS [RFC3467]) or in the transport layer (such as using ALTO [RFC5693]). The challenge is to balance higher-layer decisions about which application layer resources to use with information from the lower layers about the availability and load of network resources.

* Scalability - There are many scaling concerns that pose critical challenges to the Internet. Not least among these challenges is the size of the routing tables that routers in an IP network must maintain. As the number of devices attached to the network grows, so the number of addresses in use also grows, and because of the schemes used to assign address prefixes, the mobility of devices, and the various connectivity options between networks, the routing table sizes also grow, even more so when prefixes are not always amenable to aggregation. This problem is exacerbated by some

services (such as those supported by the IoT where several thousands of objects/sensors may be networked), where, as more devices are added to the network, the size of the routing table may affect the operation of certain routing protocols. It may be noted that scaling issues are also exacerbated by multihoming practices if a host that is multihomed is allocated a different address for each point of attachment.

* Manageability, Maintainability, and Extensibility - Operational manageability is a key requirement for network technologies: network operators must be able to determine the status of their network and understand the causes of any disruptions or problems. Further, it must be possible to maintain the networks and the technologies running in them without disrupting the services being delivered by the networks. Additionally, the network technologies developed and deployed need to be extensible so that new features can be added and new services supported without the need to invent whole new technologies.

* Security - Issues of security and privacy have been largely overlooked by the routing systems. However, there is increasing concern that attacks on routing systems can not only be disruptive (for example, causing traffic to be dropped), but may cause traffic to be redirected to inspection points that can breach the security or privacy of the payloads.

Some of the challenges outlined here were previously considered within the IETF by the IAB's "Routing and Addressing Workshop" held in Amsterdam, The Netherlands on October 18-19, 2006 [RFC4984]. Several architectures and protocols have since been developed and worked on within and outside the IETF, and these are examined in [I-D.king-irtf-semantic-routing-survey].

3. What is Semantic Routing?

Semantic Routing is the term applied to routing in an IP network that relies upon additional information to feed the route computation process, to enhance route selection decisions, and to direct the forwarding process. In addition to the routable part of the destination IP address (the prefix), such information may be present in other fields in the packet (chiefly the packet header) and configured or programmed into the routers/forwarders. Semantic Routing includes mechanisms such as "Preferential Routing", "Policy-based Routing", and "Flow steering".

In Semantic Routing, a packet forwarding engine may examine a variety of fields in a packet and match them against forwarding instructions. Those forwarding instructions may be installed by routing protocols,

configured through management protocols or a software defined networking (SDN) controller, or derived by a software component on the router that considers network conditions and traffic loads. The packet fields concerned may be the fields of an IP header, those same fields but with additional semantics, elements of the packet payload, or new fields defined for inclusion in the packet header or as a "shim" between the header and payload. In the case of additional semantics included in existing packet header fields, the approach implies some "overloading" of those fields to include meaning beyond the original definition. In all cases, a well-known definition of the encoding of the additional information is required to enable consistent interpretation within the network.

A more detailed description of Semantic routing can be found in [I-D.farrel-irtf-introduction-to-semantic-routing] and a survey of Semantic Routing proposals and research projects can be found in [I-D.king-irtf-semantic-routing-survey].

Many technical challenges exist for Semantic Routing in IP networks depending on which approach is taken. These challenges include (but are not limited to):

* The continual growth of routing tables.

* Convergence times for large networks.

* Granularity of routing decisions.

* Address consumption caused by lower address utility rate. The wastage mainly comes from aligning finite allocation for semantic address blocks.

* Encoding too many semantics into prefixes will require evaluation of which to prioritize.

* Risk of privacy/information leakage.

* Lack of visibility of the Semantic Routing information when end-to-end or edge-to-edge encryption is used.

* Burdening the user, application, or prefix assignment node.

* Source address spoofing prevention mechanisms are required.

* Overloading of routing protocols causing stability and scaling problems.

* Depending on encoding mechanisms, there may be challenges for data planes to scale the processes of finding, reading, and looking up semantic data in order to forward packets at line speed.

* Backwards compatibility with existing IP networking and routing protocols.

* Extensibility to support additional functions in the future.

* Manageability and network diagnostics to be able to determine how the network is functioning and to isolate the causes of any problems.

3.1. Architectural Considerations

Semantic data may be taken into account to integrate with existing routing architectures. An overlay can be built such that Semantic Routing is used to forward traffic between nodes in the overlay, but regular IP is used in the underlay. The application of semantics may also be constrained to within a limited domain. In some cases, such a domain will use IP, but be disconnected from the Internet. In other cases, traffic from within the domain is exchanged with other domains that are connected together across an IP network using tunnels or via application gateways. And in still another case traffic from the domain is forwarded across the Internet to other nodes and this requires backward-compatible routing approaches.

Isolated Domains: Some IP network domains are entirely isolated from the Internet and other IP networks. In these cases, packets cannot "escape" from the isolated domain into external networks and so the Semantic Routing schemes applied within the domain can have no detrimental effects on external domains. Thus, the challenges are limited to enabling the desired function within the domain.

Bridged Domains: In some deployments, it will be desirable to connect together multiple isolated domains to build a larger network. These domains may be connected (or bridged) over an IP network or even over the Internet, possibly using tunnels. An alternative to tunneling is achieved using gateway functionality where packets from a domain are mapped at the domain boundary to produce regular IP packets that are sent across the IP network.

Semantic Prefix Domains: A semantic prefix domain is a portion of the Internet over which a consistent set of semantic-based policies are administered in a coordinated fashion. This is achieved by assigning a routable address prefix (or a set of prefixes) for use with Semantic Routing so that packets may be

forwarded through the regular IP network (or the Internet). Once delivered to the semantic prefix domain, a packet can be subjected to whatever Semantic Routing is enabled in the domain.

Further discussion of architectures for Semantic Routing can be found in [I-D.farrel-irtf-introduction-to-semantic-routing].

4. Challenges for Internet Routing Research

It may not be possible to embrace all emerging scenarios with a single approach or solution. Requirements such as 5G mobility, near-space-networking, and networking for outer-space (inter-planetary networking), may need to be handled using different network technologies. Improving IP network capabilities and capacity to scale, and address a set of growing requirements presents significant research challenges, and will require contributions from the networking research community. Solutions need to be both economically feasible and have the support of the networking equipment vendors as well as the network operators.

4.1. Research Principles

Research into Semantic Routing should be founded on regular scientific research principles [royalsoc]. Given the importance of the Internet today, it is critical that research is targeted, rigorous, and reproducible.

The most valuable research will go beyond an initial hypothesis, a report of the work done, and the results observed. Although that is a required foundation, networking research needs to be independently reproducible so that claims can be verified or falsified. Further, the networks on which the research is carried out need to both reflect the characteristics that are being explicitly tested, and reproduce the variety of real networks that constitute the Internet.

Thus, when conducting experiments and research to address the questions in Section 4.2, attention should be given to how the work is documented and how meaningful the test environment is, with a strong emphasis on making it possible for others to reproduce and validate the work.

4.2. Routing Research Questions to be Addressed

As research into the scenarios and possible uses of Semantic Routing progresses, a number of questions need to be answered. These questions go beyond "Why do we need this function?" and "What could we achieve by carrying additional semantics in an IP address?" The questions are also distinct from issues of how the additional semantics can be encoded within an IP address. All of those issues are, of course, important considerations in the debate about Semantic Routing, but they form only part of the essential groundwork of research into Semantic Routing itself.

This section sets out some of the concerns about how the wider the use of Semantic Routing might impact a routing system. These questions need to be answered in separate research work or folded into the discussion of each Semantic Routing proposal.

1. What is the scope of the Semantic Routing proposal? This question may lead to various answers:

   Global: It is intended to apply to all uses of IP.

   Backbone: It is intended to apply to IP network connectivity.

   Overlay: It is to be used as an overlay network using tunneling over IP or other underlay technologies.

   Gateway: The Semantic Routing will be used within a specific domain, and communications with the wider Internet will be handled by IP and probably application gateways.

   Domain: The use of the Semantic Routing is strictly limited to within a domain or private network.

   Underlying this question is a broader question about the boundaries of the use of IP, and the limit of "the Internet". If a limited domain is used, is it a semantic prefix domain [RFC8799] where a part of the IP address space identifies the domain so that an address is routable to the domain, but the additional semantics are used only within the domain, or is the address used exclusively within the domain so that the external impact of the routability of the address and the additional semantics is not important?

2. What will be the impact on existing routing systems? What would happen if a packet carrying additional semantics was subjected to normal routing operations? How would the existing routing systems react if such a packet escaped (accidentally or

maliciously) from the planned scope of the proposal? For example: how are the semantic parts of an address distinguished from the routable parts (if, indeed, they are separable)?; is there an impact on the size and maintenance of routing tables due to the addition of semantics?; how are cryptographically generated addresses (such as [RFC3972]) made routable and kept simple enough for management?.

3. What path characteristics are needed to describe the desired paths and as input to route computation? Since one of the implications of adding semantics to IP packets is to cause special processing by routers, it is important to understand what behaviors are wanted. Such path characteristics include (but are not limited to):

   Quality: Expressed in terms of throughput, latency, jitter, drop precedence, etc.

   Resilience: Expressed in terms of survival of network failures and delivery guarantees.

   Destination: How is a destination address to be interpreted if it encodes a choice of actual destinations? Can traffic be forwarded over multiple distinct paths if multiple destination addresses are encoded?

   Security: What choices of path reduce the vulnerability of the traffic to security or privacy attacks?

   In these cases, how do the routers utilize the additional semantics to determine the desired characteristics? Or are such characteristics used to feed the route computation logic, for example, by means of metrics? What additional information about the network do the routing protocols need to gather? What changes to the routing algorithm are needed to deliver packets according to the desired characteristics? How can routes be computed with characteristics that accommodate traffic patterns, requirements, and constraints?

4. Can we solve these routing challenges with existing routing tools and methods? We can break this question into a set of more detailed questions.

* Is new hardware needed? Existing deployed hardware has certain assumptions about how forwarding is carried out based on IP addresses and routing tables. But hardware is increasingly programmable so that it may be possible to instruct the forwarding components to act on a variety of elements of the packets.

* Do we need new routing protocols? We might ask some subsidiary questions:

  - Can we make do with existing protocols, possibly by tuning configuration parameters or using them out of the box?

  - Can we make backwards-compatible modifications to existing protocols such that they work equally for today's IP addresses or addresses with extra semantics?

  - Do we need entirely new protocols or radical evolutions of existing protocols in order to enforce advanced Semantic Routing policies?

  - Should we focus on the benefits of routing solutions that are optimized for specific environments (network topologies, technologies, use cases), or should we attempt to generalize to enable wider applicability?

5. Do we need new management tools and techniques? How practical is it to debug and operate the routing system? Management of the routing system (especially diagnostic management) is a crucial and often neglected part of the problem space. A critical part of this issue is how packets within the network can be inspected by diagnostic tools (or human operators) and mapped to the routing and forwarding decisions that were made within the network in order to understand the actions made at and by upstream routers.

6. What is the impact of Semantic Routing on the security of the routing system?

   * Does the introduction of Semantic Routing provide a greater attack surface?

   * Can Semantic Routing provide greater opportunities for security by fine-grain forwarding of flows to be inspected by different security functions?

* Can Semantic Routing improve security and privacy by obscuring information in the packets, or does the inclusion of additional information risk compromising security and privacy?

* To what extent does deployment within a limited domain strengthen security or make it less of a concern?

* Does the use of Semantic Routing make it easier or harder to impose censorship, prohibit access to the Internet by specific parties, or block access to certain resources or types of service?

7. What is the scalability impact of Semantic Routing on routing systems? Scalability can be measured as:

   * Routing table size. How many entries need to be maintained in the routing tables by different routers serving different roles in the network? Some approaches to Semantic Routing may be explicitly intended to address this problem.

   * Forwarding table size. The size of the forwarding table may be less of an issue considering modern hardware, however the more granular the routing/forwarding decisions made in a router, the greater the size of this table. The size of the forwarding table has implications for memory in the forwarding engine, but also for the lookup time for forwarding each packet.

   * Routing performance. Routing performance may be considered in terms of the volume of data that has to be exchanged both to construct and maintain the routing tables at the participating routers. It may also be measured in terms of how much processing is required to compute new routes when there is a change in the network.

   * Routing convergence. This is the time that it takes for a routing protocol to discover changes (especially faults) in the network, to distribute the information about any changes to its peers, and to reach a stable state across the network such that packets are forwarded consistently.

   For all questions about routing scalability, research that presents figures based on credible example networks is highly desirable. Similar questions may be asked about the amount of forwarding state that has to be maintained in the routers.

8. To what extent can Semantic Routing be applied to multicast transmission schemes:

   * Can Semantic Routing facilitate the computation and the establishment of (service-inferred) multicast distribution trees?

   * Can specific semantics be carried in multicast addresses?

9. Is the approach extensible and maintainable? Can new features be added without increasing the complexity and in a backward compatible way? Could the approach be modified to handle evolutions in the rest of the networking infrastructure? Considerations might include the ability to encode additional options or variants within protocol fields, and the ability to add new fields. Such considerations must be actively traded against the processing overhead associated with certain encoding types.

10. What aspects need to be standardized? It is important to understand the necessity of standardization within this research. What degree of interoperability is expected between devices and networks? Is a given domain so constrained (for example, to a single equipment vendor) that standardization would be meaningless? Is the application so narrow (for example, in niche hardware environments) such that interoperability is best handled by agreements among small groups of vendors such as in industry consortia?

5. Security and Privacy Considerations

Research into Semantic Routing must give full consideration to the security and privacy issues that are introduced by these mechanisms. Placing additional information into packet header fields might reveal details of what the packet is for, what function the user is performing, who the user is, etc. Furthermore, in-flight modification of the additional information might not directly change the destination of the packet, but might change how the packet is handled within the network and at the destination.

It should also be considered how packet encryption techniques that are increasingly popular for end-to-end or edge-to-edge security may obscure the semantic information carried in some fields of the packet header or found deeper in the packet. This may render some semantic routing techniques impractical and may dictate other methods of carrying the necessary information to enable Semantic Routing.

6. IANA Considerations

This document makes no requests for IANA action.

7. Acknowledgements

Thanks to Stewart Bryant for useful conversations. Luigi Iannone, Robert Raszuk, Dirk Trossen, Ron Bonica, Marie-Jose Montpetit, Yizhou Li, Toerless Eckert, Tony Li, Joel Halpern, Stephen Farrell, Carsten Bormann, David Hutchison, Jeffery He, Dino Farinacci, Greg Mirsky, and Jeff Haas made helpful suggestions.

This work is partially supported by the European Commission under Horizon 2020 grant agreement number 101015857 Secured autonomic traffic management for a Tera of SDN flows (Teraflow).

8. Contributors

Joanna Dang
Email: dangjuanna@huawei.com

9. Informative References

[I-D.farrel-irtf-introduction-to-semantic-routing]
Farrel, A. and D. King, "An Introduction to Semantic Routing", Work in Progress, Internet-Draft, draft-farrel-irtf-introduction-to-semantic-routing-03, 22 January 2022, <https://www.ietf.org/archive/id/draft-farrel-irtf-introduction-to-semantic-routing-03.txt>.

[I-D.king-irtf-semantic-routing-survey]
King, D. and A. Farrel, "A Survey of Semantic Internet Routing Techniques", Work in Progress, Internet-Draft, draft-king-irtf-semantic-routing-survey-03, 26 November 2021, <https://www.ietf.org/archive/id/draft-king-irtf-semantic-routing-survey-03.txt>.

[RFC3467] Klensin, J., "Role of the Domain Name System (DNS)", RFC 3467, DOI 10.17487/RFC3467, February 2003, <https://www.rfc-editor.org/info/rfc3467>.

[RFC3972] Aura, T., "Cryptographically Generated Addresses (CGA)", RFC 3972, DOI 10.17487/RFC3972, March 2005, <https://www.rfc-editor.org/info/rfc3972>.

[RFC4984] Meyer, D., Ed., Zhang, L., Ed., and K. Fall, Ed., "Report from the IAB Workshop on Routing and Addressing", RFC 4984, DOI 10.17487/RFC4984, September 2007, <https://www.rfc-editor.org/info/rfc4984>.

[RFC5693] Seedorf, J. and E. Burger, "Application-Layer Traffic Optimization (ALTO) Problem Statement", RFC 5693, DOI 10.17487/RFC5693, October 2009, <https://www.rfc-editor.org/info/rfc5693>.

[RFC8799] Carpenter, B. and B. Liu, "Limited Domains and Internet Protocols", RFC 8799, DOI 10.17487/RFC8799, July 2020, <https://www.rfc-editor.org/info/rfc8799>.

[royalsoc] The Royal Society, "Evidence synthesis : Principles", Web page, Principles for good evidence synthesis, 19 September 2018, <https://royalsociety.org/topics-policy/projects/evidence-synthesis/principles/>.

Authors' Addresses

Daniel King
Lancaster University
United Kingdom
Email: d.king@lancaster.ac.uk

Adrian Farrel
Old Dog Consulting
United Kingdom
Email: adrian@olddog.co.uk

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

IRTF
Internet-Draft
Intended status: Informational
Expires: 1 December 2022

D. King
Lancaster University
A. Farrel
Old Dog Consulting

30 May 2022

A Survey of Semantic Internet Routing Techniques
draft-king-irtf-semantic-routing-survey-04

Abstract

The Internet Protocol (IP) has become the global standard in any computer network, independent of the connectivity to the Internet. Generally, an IP address is used to identify an interface on a network device. Routing protocols are also required and developed based on the assumption that a destination address has this semantic with routing decisions made on addresses and additional fields in the packet headers.

Over time, routing decisions were enhanced to route packets according to additional information carried within the packets and dependent on policy coded in, configured at, or signaled to the routers. Many proposals have been made to add semantics to IP addresses. The intent is usually to facilitate routing decisions based solely on the address and without finding and processing information carried in other fields within the packets.

This document is presented as a survey to support the study and further research into clarifying and understanding the issues. It does not pass comment on the advisability or practicality of any of the proposals and does not define any technical solutions

This Internet-Draft will expire on 1 December 2022.

Copyright Notice

Table of Contents

1. Introduction

The Internet continues to expand rapidly, and the Internet Protocol (IP) has become the global standard in many types of computer network independent of whether or what connectivity to the Internet it has. At the same time, there are increasingly varied expectations of the services and service level objectives that can be required from networks. For example, packet-delivery quality expectations beyond best effort is a growth area: throughput, latency, error recovery, and (absence of) packet or connectivity loss, reordering, or jitter. Requirements include relative or absolute guarantees or predictable elastic changes under contention on these performance factors. This places significant pressure on Service Providers to be aware of the type of services being delivered, and to have access to sufficient information about how individual packets should be treated to meet the user, application and application instance requirements.

IP addresses facilitate the identification of how a device is attached to the Internet and how it is distinguished from every other device. Addresses are used to direct packets to a destination (destination address) and indicate to where the receiver and network replies and error messages should be sent (source address). An IP address may be assigned to each network interface of a device

connected to a network that uses IP. Applications use IP addresses to both identify a host and to indicate the physical or virtual location of the host.

This document presents a brief survey of proposals to extend the semantics of IP addresses by assigning additional meanings to some parts of the address, or by partitioning the address into a set of subfields that give scoped addressing instructions. Some of these proposals are intended to be deployed in limited domains [RFC8799] that are IP-based, while other proposals are intended for use across the Internet. Limited domains may present their own challenges in terms of ensuring the perimeter of the domains, and connecting domains across the Internet.

The impact that some proposals may have on routing systems could require clean-slate solutions, hybrid solutions, extensions to existing routing protocols, or potentially no changes at all. A separate document ([I-D.king-irtf-challenges-in-routing]) describes the challenges to the routing system presented by changes to IP address semantics, and sets out research questions that should be investigated by those proposing new semantic address schemes.

2. Network Path Selection

Two approaches are typically used for network path selection. Firstly, a priori assessment by having the feasible paths and constraints computed in advance. Secondly, real-time computation in response to changing network conditions.

The first approach may be conducted offline and allows for concurrent or global optimization based on constraints and policy. However. as network size and complexity increase, the required computing power may increase exponentially for this type of computation.

The second approach must consider the speed of calculation where complex constraints are applied to the path selection. This processing may delay service setup and the responsiveness to changes (such as failures) in the network. Network topology filters may be applied to reduce the complexity of the network data and the computation algorithm, however, the path computation accuracy and optimality may be negatively affected.

In both approaches, the amount of information that needs to be imported and processed can become very large (e.g., in large networks, with many possible paths and route metrics), which might impede the scalability of either method both in terms of the storage and the distribution of the information.

In the last decade, significant research has been conducted into the architecture of the future Internet (for example, [RESEARCHFIAref] and [ITUNET2030ref]). During this research, several techniques emerged, highlighting the benefits of path awareness and path selection for end-hosts, and multiple path-aware network architectures have been proposed, including SCION [SCIONref] and RINA [RINAref], and the work of the Path Aware Networking Research Group (PANRG) as discussed later in this document.

When choosing the best paths or topology structures, the following may need to be considered:

* The method by which a path, or set of paths, is to be calculated. For example, a path may be selected automatically by the routing protocol or imposed (perhaps for traffic engineering reasons) by a central controller or management entity.

* The criteria used for selecting the best path. For example, classic route preference, or administrative policies such as economic costs, resilience, security, and if requested, applying geopolitical considerations.

2.1. Path Aware Routing

The current architecture for IP networking is built using a best-effort philosophy. There are techniques discussed in this document that attempt better-than-best-effort delivery. The start-point and end-point of a path are identified using IP addresses, and traffic is steered along the path that does not necessarily follow the "shortest path first" route through the network. Furthermore, the path might not run all the way from a packet’s source to its destination. The assumption is that a packet reaching the end of a path is forwarded to its destination using best-effort techniques.

Evaluating and building paths that respect requirements beyond the simple best-effort model is particularly challenging and computationally heavy since numerous quality-related parameters need to be considered.

3. What is Semantic Routing?

Networks are often divided into addressing regions for various administrative or technological reasons. Different routing paradigms may be applied in each region, and specific "private" semantics may be applied to the IP addresses within a single region

These address semantics are established using customer types, customer connections, topological constraints, performance groups, and security, etc. Service Providers or network operators will apply local policies to user and application packets as they enter the network possibly mapping addresses, or encapsulating them with an additional IP header. In some case, the packet has its source and destination within a single network and the network operator can apply address semantics policies across the whole network. In other cases (such as general IP-based traffic), a packet will require a path across multiple networks, and each may apply its own set of traffic forwarding policies. In these cases, there is often no consistency or guaranteed performance unless a Service Level Agreement (SLA) is applied to traffic traversing multiple networks.

Semantic routing proposals may apply to addresses in a specific domain, or domain set. In this context, a "limited domain" means that the interpretation of the address, in a semantic routing domain, is only applicable to a well-defined set of network nodes, or specific points in the network. If a packet bearing an address with a modified semantic were to escape from the domain, the special meaning of the address would be lost. Additionally (or alternatively), the meaning of "specific points in the network" may be applied to the source and destination nodes of a packet, while all transit nodes are unaware of the special semantic. However, it could be the case that some key transit nodes are able to access the meaning of the address and so apply special routing or other functions to the packet.

Such proposals include the following:

* Providing semantics specific to mobile networks so that a user or device may move through the network without disruption to their service [CONTENT-RTG-MOBILEref].

* Enabling optimized multicast traffic distribution by encoding multicast tree and replication instructions within addresses [MULTICAST-SRref].

* Using addresses to identify different device types so that their traffic may be handled differently [SEMANTICRTG].

* Content-based routing (CBR), forwarding of the packet based on message content rather than the destination addresses [OPENSRNref].

* Deriving IP addresses from the lower layer identifiers and using addresses depending on the underlying connectivity (for example, [RFC6282].

* Identifying hierarchical connectivity so that routing can be simplified [EIBPref].

* Providing geographic location information within addresses [GEO-IPref].

* Indicating the application or network function on a destination device or at a specific location; or enable Service Function Chaining (SFC).

* Expressing how a packet should be handled, prioritized, or allocated network resources as it is forwarded through the network [TERASTREAMref].

* Using cryptographic algorithms to mask the identity of the source or destination, masking routing tables within the domain, while still enabling packet forwarding across the network [BLIND-FORWARDINGref].

In many cases, it may be argued that existing mechanisms applied on top of the common address semantic defined in [RFC4291] can deliver the correct functionality for these scenarios. That is, packets may be tunneled over IP using several existing encapsulation techniques. Nevertheless, there is pressure to reduce the amount of encapsulation (partly to resist reduction in the maximum transmission unit (MTU) over the network, and partly to achieve a flatter and more transparent network architecture). This leads to investigations into whether the current IP addresses can be "overloaded" (without any negative connotations being attached to that word) by adding semantics to the addresses.

Semantic Routing is the process of routing packets that contain IP addresses with additional semantics, possibly using that information to perform policy-based routing or other enhanced routing functions. Thus, faciliating enhanced routing decisions based on these additional semantics and provide differentiated paths for different packet flows, distinct from simple shortest path first routing. The process of known as Semantic Routing is discussed further in [I-D.farrel-irtf-introduction-to-semantic-routing].

Key use cases exist for semantic routing, typically for specific applications and deplyments, including low earth orbit (LEO) satellite constellations [I-D.lhan-satellite-semantic-addressing].

Based on a variety of use cases, key technical challenges exist for semantic routing: these are discussed further in [I-D.king-irtf-challenges-in-routing].

3.1. Architectural Considerations

Semantics may be applied in multiple ways to integrate with existing routing architectures. The most obvious is to build an overlay such that IP is used only to route packets between network nodes that utilize the semantics at a higher layer. There are several uses of this approach, including Service Function Chaining (SFC) (see Section 4.2.3) and Information Centric Networking (ICN) (see Section 5.4.1). An overlay may be achieved in a higher layer, or may be performed using tunneling techniques (such as IP-in-IP) to traverse the areas of the IP network that cannot parse additional semantics and so join together those nodes that use the semantics.

The application of semantics may also be constrained to within a limited domain. In some cases, such a domain will use IP, but be disconnected from Internet. In those cases, the challenges are limited to enabling the desired function within the domain. In other cases, traffic from within the domain is exchanged with other domains that are connected across an IP-based network using tunnels or via application gateways. And in another case traffic from the domain is routed across the Internet to other nodes and this requires backward-compatible routing approaches, tunnels, or gateway functions.

Limited domains [RFC8799] are a fact of networking life. They are used to safely deploy or test features and functions in a controlled environment so that they cannot contaminate other networks or the Internet in general. Examples of a limited domain in use today include:

* Internet of Things (IoT) networks such as factory floors or home networks

* Deterministic Networks (DetNet) that operate in campus networks or private WANs to provide deterministic data paths with bounded latency, low loss, predictable jitter, and high reliability.

* Content Distribution Networks (CDN) where clusters of servers share content provision but may also need to be interconnected.

* Physical security may be provided for a site simply by not permitting traffic to enter or leave the site. This may be expanded by connecting multiple sites together using tunnels across the Internet to form a Virtual Private Network (VPN).

Limited domains are also used as a driver for innovation. They provide a safe space to run experiments and deploy new functions such as advances in traffic steering, improvements in security, and new routing protocols. A limited domain is a way to achieve incremental

deployability on an isolated island, and this enables innovation that may (or may not) percolate to the whole Internet at a later stage. For example, experiments to increase the programmability of network forwarding functions need to be carried out in networks of similarly capable nodes (to avoid the risks of broken interoperability or forwarding loops), yet these experiments need to use real user data that is flowing between hosts and servers.

Because limited domains don’t always operate in isolation they may need to be connected to other domains over the Internet, or other nodes within the wider Internet.

4. Existing Approaches for Routing Based on Additional Semantics

Several IETF-based approaches are available to allow service providers to perform policy-based routing, including identifying and marking IP traffic either by changing the semantic of IP addresses or by adding such a semantic in other fields/namespaces, enabling differentiated handling by transit routers (queuing, dropping, forwarding, etc.). The sections below distinguish between those schemes that perform routing based on information other than IP addresses, those that establish an overlay network in which to apply semantics, and those that add semantics to the addresses. A further separate group of approaches is presented here to cover the concept of group semantics where a single address identifies more than one endpoint.

4.1. Non-Address-Based Routing

Many routing schemes examine the destination address field and other fields in the packet header to make routing decisions. These approaches (sometimes referred to as "policy-based routing") allow packets to follow different paths through the network depending on semantics assigned to these other fields or based on hashing algorithms operating on the values of those fields.

4.1.1. Deep Packet Inspection

Deep Packet Inspection (DPI) may be used by a router to learn the characteristics of packets in order to forward them differently. This involves looking into the packet beyond the top-level network-layer header to identify the payload. Once identified, the traffic type can be used as an input for marking the packets for network handling, or for performing specific policies on the packets.

However, DPI may be expensive both in processing costs and latency. The processing costs means that dedicated infrastructure is necessary to carry out the function, and this may have an associated financial

cost. The latency incurred may be too much for use with any delay/jitter sensitive applications. As a result, DPI is difficult for large-scale deployment and its usage is often limited to specific functions at the edge of the network.

Despite this, "shallow DPI" is commonplace in routers today as they examine the five-tuple of source address, destination address, payload protocol, source port, and destination port to perform a hash function for ECMP purposes (a form of policy-based routing).

4.1.2. Differentiated Services

Quality of Service (QoS) based on Differentiated Services [RFC2474] is a widely deployed framework specifying a simple and scalable coarse-grained mechanism for classifying and managing network traffic. However, in a service providers network, DiffServ codepoint (DSCP) values cannot be trusted when they are set by the customer, and may have different meanings as packets are passed between networks.

In real-world scenarios, Service Providers deploy "remarking" points at the edges of their network, re-classifying received packets by rewriting the DSCP field according to local policy using information such as the source/destination address, IP protocol number, transport layer source/destination ports, and possibly applying DPI as described in Section 4.1.1.

The traffic classification process and node-by-node processing leads to increased packet processing overhead and complexity at the edge of the Service Providers network.

4.1.3. IPv6 Extension Headers

[RFC8200] defines the IPv6 header and also a number of extension headers. These extension headers can be used to carry additional information that may be used by transit routers (the hop-by-hop options header) or by the destination identified by the destination address field (the destination options header). In addition, these extension headers could encode additional semantics that might enable routing decisions and determine what functions and operations should be performed on a packet.

[RFC7872] and [I-D.ietf-v6ops-ipv6-ehs-packet-drops] provide some discussions about the operational problems of using IPv6 extension headers, especially in multi-domain environments, while [I-D.bonica-6man-ext-hdr-update] proposes to update RFC 8200 with guidance regarding the processing, insertion and deletion of IPv6 extension headers.

4.2. Semantic Overlays

An overlay network is built on top of an underlay or transport network. Packets are encapsulated with the header for the overlay network to carry the additional information needed to provide the desired function, and then the packets are encapsulated for transport through the underlay network. In this case, no changes are made to the meaning of the IP addresses in the underlay, but the destination address identifies the next hop in the overlay network rather than the ultimate destination of the packet. In this way, packets can be steered through different overlay nodes where routing decisions can be made.

4.2.1. Application-Layer Traffic Optimization

Application-Layer Traffic Optimization (ALTO) [RFC7285] is an architecture and protocol. ALTO defines abstractions and services to provide simplified network views and network services to guide the application usage of network resources, including cost.

An ALTO server gathers information about the network and answers queries from an ALTO client that wants to find a suitable path for traffic. ALTO responses are typically used to route whole flows (not individual packets) either to suitable destinations (such as network functions) or onto paths that have specific qualities.

4.2.2. Multipath TCP

Multipath TCP (MPTCP) [RFC8684] enables the use of TCP in a multipath network using multiple host addresses. A Multipath TCP connection provides a bidirectional bytestream between two hosts communicating like normal TCP and thus does not require any change to the applications. However, Multipath TCP enables the hosts to use different paths with different IP addresses to exchange packets belonging to the MPTCP connection.

MPTCP increases the available bandwidth, and so provides shorter delays; it increases fault tolerance, by allowing the use of other routes when one or more routes become unavailable; and it enables traffic engineering and load balancing.

4.2.3. Service Function Chaining

Service Function Chaining (SFC) [RFC7665] is the process of sending traffic through an ordered set (a sequence) of abstract service functions. This may be achieved using an overlay encapsulation such as the Network Service Header (NSH) [RFC8300] or MPLS [RFC8596] that rely on tunneling through an underlay without any additional semantics applied to the IP addresses.

Alternatively, SFC can be performed by adding semantics to the addresses, for example, as in Section 4.3.3.

4.2.4. Path Computation Element

The Path Computation Element (PCE) [RFC4655] is an architecture and protocol [RFC5440] that can be used to assist with network path selection. A PCE is an entity capable of computing paths for a single or set of services. A PCE might be a network node, network management station, or dedicated computational platform that is resource-aware and has the ability to consider multiple constraints for sophisticated path computation. PCE applications compute label switched paths for MPLS and GMPLS traffic engineering, but the PCE has been extended for a variety of additional traffic engineering problems.

4.3. Semantic Routing

In semantic routing, additional information or meaning is placed into the IP address, and this is used to route packets within the network.

4.3.1. Locator/ID Separation Protocol (LISP)

The Locator/ID Separation Protocol (LISP) [RFC6830] was published by the IETF as an Experimental RFC in 2013 and is now being moved to the Standards Track [I-D.ietf-lisp-rfc6830bis] and [I-D.ietf-lisp-rfc6833bis]. LISP separates IP addresses into two numbering spaces: Endpoint Identifiers (EIDs) and Routing Locators (RLOCs). The former, the EIDs, are used to identify communication end-points (as the name states) as well as local routing and forwarding in the edge network. The latter, RLOCs, are used to locate the EIDs in the Internet topology end are usually the address of ASBRs (Autonomous System Border Routers). IP packets addressed with EIDs are encapsulated with RLOCs for routing and forwarding over the Internet.

As end-to-end packet forwarding includes both EIDs and RLOCs an additional control-plane is needed. This control plane provides a mapping system and basic traffic engineering capabilities. Multihoming becomes easier because one EID can be associated to more than one RLOC or even to a local network address prefix.

4.3.2. Identifier-Locator Network Protocol

The Identifier-Locator Network Protocol (ILNP) [RFC6740] is an experimental network protocol designed to separate the two functions of network addresses: identification of network endpoints, topology or location information. Differently from LISP, ILNP encodes both locator and identifier in the IPv6 address format (128 bits). More specifically, the most significant 64 bits of the 128 bits IPv6 address is the locator, while the less significant 64 bits form the identifier. Upon reaching the destination network, a cache is used to find the corresponding node. Furthermore, DNS can be dynamically updated, which is essential for mobility and also for provider-independent addresses. Similar to LISP, multihoming can be set by assigning multiple locators to the same identifier. In addition, identifiers can also be encrypted for privacy reasons. It was intended that ILNP should be backwards-compatible with existing IP, and that it should be incrementally deployable.

4.3.3. Segment Routing

Segment Routing (SR) [RFC8402] leverages the source routing paradigm. A node steers a packet through an ordered list of instructions, called "segments". A segment can represent any instruction, topological or service based. A segment can have a semantic local to an SR node or global within an SR domain. SR provides a mechanism that allows a flow to be restricted to a specific topological path, while maintaining per-flow state only at the ingress node(s) to the SR domain.

In SR for IPv6 networks (SRv6) segment routing functions are used to achieve a networking objective that goes beyond packet routing, in order to provide "network programming" [RFC8986]. The network program is expressed as a list of instructions, which are represented as 128-bit segments, called Segment Identifiers (SID) - encoded and presented in the form of an IPv6 address. The first instruction of the network program is placed in the Destination Address field of the packet. If the network program requires more than one instruction, the remaining list of instructions is placed in the Segment Routing Extension Header (SRH)[RFC8754].

An SRv6 instruction can represent any topological or service-based instruction. The SRv6 domain is the service provider domain where SRv6 services are built to transport any kind of customer traffic including IPv4, IPv6, or frames. SRv6 is the instantiation of Segment Routing deployed on the IPv6 data plane. Therefore, in order to support SRv6, the network must first be enabled for IPv6.

The SRH in the IPv6 header is only processed for nodes forwarding traffic if the destination address identifies the local node. In this case, the node must take several actions, including reading the SRH, performing any node-specific actions identified by the destination address or the next SIDs in the SRH, and re-writing the IPv6 destination address field using information from the SRH before forwarding the packet.

4.3.4. Preferred Path Routing

Preferred Path Routing (PPR) [I-D.chunduri-lsr-isis-preferred-path-routing] is a proposed routing protocol mechanism where alternate forwarding state is installed for a set of different preferred paths. Each preferred path is described as an ordered linear list of nodes, links, and network functions, and the path is identified by a network-global preferred path identifier. If a packet is marked with preferred path identifier, it is forwarded according to the preferred path that has been installed on the router. If a packet is not marked or if the preferred path is not installed on the router, the packet is forwarded using the normal shortest path first algorithm.

In PPR, the preferred path identifier is encoded in an IP address, but the address is only used in an encapsulation of the end-to-end packet. This approach is a hybrid in that it is applying a different meaning to the IP addresses, using that meaning in an encapsulation, but routing the packets through an existing IP network.

4.3.5. Connectionless Network Protocol

The Connectionless Network Protocol (CLNP) [CLNPref] is a network layer encoding that supports variable length, hierarchical addressing. It is widely deployed in many communications networks and is the ITU-T’s standardized encoding for packets in the management plane for Synchronous Digital Hierarchy (SDH) networks. For a while, CLNP was considered in competition with IP as the network layer encoding for the Internet, but IP (in conjunction with TCP) won out.

Many of the considerations for semantic addressing can be handled using CLNP, and it is particularly well suited to applications that demand variable-length addresses or that structure addresses hierarchically for routing or geo-political reasons.

Routing for CLNP can be achieved using the IS-IS routing protocol in its full form as documented in [ISISref] rather than its IP-only form [RFC1195]. While this may make it possible to use CLNP alongside IP in some routed networks, it does not integrate the use of IP addresses with additional semantics with the historic use of IP addresses except in "ships that pass in the night" fashion. Alternatively, [RFC1069] explains how to carry regular IP addresses in CLNP.

4.4. Group Semantics

A mayor enhanced addressing semantic in IP is called "group semantics". Here, an IP address identifies more than one individual interface or node. This facilitates the delivery of a packet to any one of a group of destinations, or to all group members.

4.4.1. Multicast

Multicast address semantics support delivery to all members of a group of destinations. This is a controlled variant of broadcasting where packets are delivered to all possible receivers in a particular (static) scope such as a multi-access link. Membership of a multicast link is dynamically signalled by the group members, and a group is identified by a specific address.

IP multicast [RFC1112], based on the protocol and service definition aspects of Steve Deering’s PhD, is widely deployed for IPv4. It is equally adopted and used in IPv6 using the addressing architecture specified in [RFC4291]. In IP multicast (Any Source Multicast - ASM) any node can send to the multicast group and have its packets delivered to all members of the group.

Research deployments in the 1990s (the so called 'MBone' [MBONEref]) indicated that IP multicast gave rise to a number of issues related to address assignment, implementation, scale, and security. The problem of allocation and management of IP multicast (group) addresses led to several proposals, including Multicast Address Dynamic Client Allocation Protocol (MADCAP) [RFC2730], the Multicast Address Allocation Architecture (MALLOC) [RFC6308], the Multicast Address-Set Claim Protocol (MASC) [RFC2909], and the Multicast-Scope Zone Announcement Protocol (MZAP) [RFC2776], but none was widely adopted. Attempts to create a complete routing protocol suite for IP multicast service model within the IETF resulted in the Multicast Source Discovery Protocol (MSDP) being published as an experimental RFC [RFC3618].

The popularity of multicast as a concept and the widespread deployment of commercial IPv4 multicast led to the development of "Source Specific Multicast" service (SSM) [RFC4607]. In SSM, the combination of the Source and Group addresses (S,G) of an IP multicast packet form a so-called SSM channel address, which identifies group of receivers and implies a single permitted sender. Receivers subscribe to every SSM channel.

From a service user's perspective, SSM solves the security issue (only valid sources can send traffic) and the address assignment issue (all group addresses are relative to the source address). For the operator, SSM also eliminates the complex operational requirements of ASM.

4.4.2. Automatic Multicast Tunneling

Automatic Multicast Tunneling (AMT) [RFC7450] is a protocol for delivering multicast traffic from sources in a multicast-enabled network to receivers that lack multicast connectivity to the source network. The protocol uses UDP encapsulation and unicast replication to provide this functionality as a hybrid solution using both multicast routing and an overlay approach.

4.4.3. Bit Index Explicit Replication

The IETF standardized or otherwise deployed protocol solutions in support of ASM and SSM in about 2015 relied all on per-hop, per ASM-group/per-SSM-channel stateful hop-by-hop forwarding/replication. Service Provider at that time were starting to removing or reduce heavy-weight control and per-hop forwarding processing in unicast caused by MPLS LDP/RSVP-TE driven designs, replacing it with more lightweight MPLS-SR and later SRv6 forwarding and associated control planes. But to reduce the cost for multicast service, the only transit-hop stateless solution available was ingress-replication,

tunnel multicast across unicast, hence trading hop-by-hop state (and its control and management plane cost) in the network against traffic overhead and (under congestion) higher latency.

Bit Index Explicit Replication (BIER) [RFC8279] addresses these problems. BIER does not contain the notion of ASM or SSM groups. Instead, a sender enumerates the set of receivers to which the packet is to be delivered. The network routers forward packets and replicate them onto the shortest paths to the destinations. As the packets are replicated, so the enumeration of the receivers is pruned on each copy of the packet.

BIER is able to use existing routing protocols without modification, but requires enhancements in the forwarding plane to encode, parse, and act on the set of receivers. The BIER information is carried in new encapsulations [RFC8296] that is carried hop-by-hop in IP. Thus, the additional semantic is in an overlay.

5. Overview of Current Routing Research Work

This section presents a limited survey of techniques and projects that provide mechanisms to facilitate path and forwarding decisions based on contextual information.

More recently, the proceedings of the June 2021 Semantic Addressing and Routing for Future Networks (SARNET-21) Workshop was compiled and published as [I-D.galis-irtf-sarnet21-report]. It captures the views and positions of the participants as expressed during the workshop.

5.1. Forwarding

Some research work is engaged in examining the emerging set of new requirements that exceed the network and transport services of the current Internet, which only delivers "best effort" service. This work aims to determine what features can be built on top of existing solutions by adding additional new components or features. A starting point for this discussion can be found in [I-D.bryant-arch-fwd-layer-ps].

Several additional techniques for improving IP-based routing have been proposed, some of these are highlighted below.

5.1.1. Path Aware Networking

The IRTF's Path Aware Networking Research Group [PANRGref] aims to support research in bringing path aware techniques into use in the Internet. This research overlaps with many past and existing IETF and IRTF efforts, including multipath transport protocols, congestion control in multiply-connected environments, traffic engineering, and alternate routing architectures.

[I-D.irtf-panrg-path-properties] offers a vocabulary of path properties. By doing so it gives some clarity of the distinction between path aware routing and semantic routing as considered in this document.

[I-D.irtf-panrg-what-not-to-do] provides a catalog and analysis of past efforts to develop and deploy Path Aware techniques. Most, but not all, of these mechanisms were considered at higher levels, although some apply at the IP routing and forwarding layer.

5.2. Trust and Accountability

5.2.1. Scalability, Control, and Isolation on Next-Generation Networks

The SCION (Scalability, Control, and Isolation on Next-Generation Networks) [SCIONref] inter-domain network architecture has been designed to address security and scalability issues and provides an alternative to current Border Gateway Protocol (BGP) solutions. The SCION proposal combines a globally distributed public key infrastructure, a way to efficiently derive symmetric keys between any network entities, and the forwarding approach of packet-carried forwarding state.

SCION End-hosts fetch viable path segments from the path server infrastructure, and construct the exact forwarding route themselves by combining those path segments. The architecture ensures that a variety of combinations among the path segments are feasible, while cryptographic protections prevent unauthorized combinations or path-segment alteration. The architecture further enables path validation, providing per-packet verifiable guarantees on the path traversed.

5.3. Layering

5.3.1. Recursive InterNetwork Architecture

Recursive Inter Network Architecture (RINA) [RINAref] builds upon the principle that applications communicate through Inter-process Communication (IPC) facilities. For an application to communicate through the distributed IPC facility, it only needs to know the name of the destination application and to use the IPC interface to request communication.

By leveraging IPC concepts, RINA allows two processes to communicate, IPC requires certain functions such as locating processes, determining permission, passing information, scheduling, and managing memory. Similarly, two applications on different end-hosts should communicate by utilizing the services of a distributed IPC facility (DIF). A DIF is an organizing structure, generally referred to as a "layer".

The scope and functions provided by the different IPC facilities may vary given the different type of network and performance goals. Moreover, an IPC layer may recursively request services from other IPC layers. The idea of recursively using multiple inter-process communication services creates a multilayer structure repeated until an IPC facility can fit well for physical technologies, e.g., wired or wireless networks.

5.4. Naming

5.4.1. Information Naming

Information-Centric Networking (ICN) [ICNref] is an approach to evolve the Internet infrastructure away from a host-centric paradigm, based on perpetual connectivity and the end-to-end principle, to a network architecture in which the focal point is information (or content or data) that is assigned specific identifiers.

Several scenarios exist for semantic-based networking, providing reachability based on Content Routing [CONTENTref] and Name Data Networking [NDNref]. The technology area of ICN is now reaching maturity, after many years of research and commercial investigation. A technical discussion into the deployment and operation of ICNs continues in the IETF: [RFC8763] provides several important deployment considerations for facilitating ICN and practical deployments.

Although ICN is primarily an overlay technology, a more recently concept, Hybrid-Information-Centric Networking (hICN), has been introduced [HICNref]. In an hICN environment the ICN aspect is integrated into the IPv6 architecture, reusing existing IPv6 packet

formats with the intention of maintaining compatibility with existing and deployed IP network technology without creating overlays that might require a new packet format or additional encapsulations. The work is described in [I-D.muscariello-intarea-hicn].

5.4.2. Service Naming

5.4.2.1. Dynamic Anycast

Dyncast (Dynamic anycast) addresses the problem of directing traffic from a client to one service instance among several available, while considering decision metrics beyond shortest path when doing so. Those service instances are therefore possible destinations for a specific service demand. [I-D.liu-dyncast-ps-usecases] outlines several use cases where such traffic steering requirement is desirable and may occur, such as in edge computing scenarios but also in distribution of video content in scenarios like autonomous driving. The draft also outlines problems with existing solutions, most notably latency in changing relations from one service instance to another due to a change in metric, which defines that decision (e.g., load in servers, latency, or a combination of several such metrics).

Key to the proposed dyncast [I-D.li-dyncast-architecture] architecture is to build on the notion of (IP) anycast, while changing the addressing semantic from a locator-based addressing to a service-oriented one. Here, the initial "service demand" packet is being identified through a service identifier as destination address. This identifier is then mapped onto a binding IP (locator-based) address at the ingress of the network, allowing for locator-based routing to be used throughout the network. The ingress-based architecture is designed in such a way that ingress nodes upon arrival of a new service demand can determine which instance (i.e., which binding IP address) to use considering both network- and service-related metrics. Furthermore, these metrics can be distributed among ingress nodes in various ways, including over a routing protocol solution.

The overall forwarding decision is based on the adherence to what is termed "instance affinity", i.e., the need to adhere to a previous routing decision for more than one packet, unlike IP forwarding on locator addresses. This affinity is created, by means of a binding table on the ingress nodes, since often more than one packet is needed for the overall service-level transaction with a specific service instance. For instance, HTTP requests may span more than one routed packet. Also, a service instance may also create ephemeral state, which requires the client to continue communicating with this instance for the duration of this state. While the affinity is

entirely defined by the application layer protocol, the network layer takes the affinity marking as input into the decision to renew its routing decision.

5.4.2.2. Prioritycast

A modification to anycast that can be instantiated by additional engineering in the routing system is called "prioritycast". Instead of relying on the shortest path forwarding semantic, prioritycast directs all traffic to the anycast address instance that is reachable and has the highest priority. This approach only requires small modifications to routing protocols so that priorities are advertised along side the addresses.

Prioritycast was originally introduced as a recommended operational practice for deployments of Bidirectional PIM (Bidir-PIM) [RFC8736] which requires a single active instance of its Rendezvous Point (RP) service. The RP is the root of a bidirectional tree and prioritycast addresses for RP allow fast failover without additional redundancy protocols beside the routing protocol, which would otherwise be necessary for such a redundancy service.

5.4.3. Structured Topological Naming

The Internet uses DNS for single-level name resolution, converting user-level domain names into IP addresses. However, techniques are being proposed for multiple levels of name resolution; these would include: application-level and user-level descriptors, service identifiers, function identifiers and endpoint identifiers, which may then be mapped to IP addresses. These additional levels of naming and resolution would allow services and components to construct the service to be easily identifiable and directly and persistently named.

5.4.4. Geographical Naming

TBD

5.4.5. Path-based Naming

TBD

5.4.5.1. ICNP

Information-centric networking (ICN) is an approach to evolve the Internet infrastructure to directly support this use by introducing uniquely named data as a core Internet principle. Data becomes independent from location, application, storage, and means of transportation, enabling in-network caching and replication.

5.4.5.2. Reed

TBD

5.4.6. Content-Based Routing

The OpenSRN [OPENSRNref] project proposed a Content-Based Routing Scheme (CBR) that uses packet content and header information to forward traffic conetextually. This proposal uses a novel software defined networking architecture to provide a semantic routing for big data network applications.

5.5. Routing

5.5.1. Inter-Domain Routing

5.5.1.1. Expedited Internet Bypass Protocol

The Expedited Internet Bypass Protocol (EIBP) [EIBPref] is a clean slate approach to routing and forwarding in the Internet using the Internet infrastructure, but bypassing the Internet Protocol (IP). The EIBP method may be deployed in current routers and when invoked for a specific end to end IP hosts or networks, EIBP bypasses the heavy traffic and security challenges faced at Layer-3. EIBP does not require routing protocols, instead it abstracts network structural (physical or logical) information into intelligent forwarding addresses that are acquired by EIBP routers automatically.

The Forwarding tables used by EIBP are proportional to the connectivity (degree) at a routing device making the protocol scalable. The EIBP routing system does not require network-wide dissemination. Topology change impacts are local and thus instabilities on topology changes are minimal. EIBP is a low configuration protocol, which can be deployed in an AS and extended to multiple ASes independently. EIBP evaluations were conducted using GENI testbeds and compared to IP using Open Shortest Path First and Border Gateway Protocol. Significant performance improvements in terms of convergence and churn rates highlight the capabilities of EIBP.

5.5.2. Intra-Domain Routing

5.6. No Changes Needed

It is entirely possible that some forms of modified address semantic will work perfectly well with existing routing protocols and mechanisms either across the whole Internet or within limited and carefully controlled domains. Claims for this sort of functionality need to be the subject of careful research and analysis as the existing protocols were developed with a different view of the meaning of IP addresses, and because routing systems are notoriously fragile.

5.7. Use Cases

Several documents are availible that discuss the requirements for applications and services that may benefit from Semantic Routing techniques, including:

o [I-D.boucadair-irtf-sdn-and-semantic-routing] This document examines the applicability of SDN techniques to Semantic Routing and provides considerations for the development of Semantic Routing solutions in the context of SDN.

o [I-D.kw-rtgwg-satellite-rtg-add-challanges] This document summerises near-to-mid-term space-networking problems; it outlines the key components, challenges, and requirements for integrating future space-based network infrastructure with existing networks and mechanisms. Furthermore, this document highlights the network control and transport interconnection, and identify the resources and functions required for successful interconnection of space-based and Earth-based Internet infrastructure.

6. Challenges for Internet Routing Research

Improving IP-based semantic network routing capabilities and capacity so that they scale and address a set of growing requirements presents significant research challenges, and will require contributions from the networking research community.

6.1. Routing Research Questions to be Addressed

As research into the scenarios and possible uses of semantic routing progresses, a number of questions need to be addressed in the scope of routing. These questions go beyond "Why do we need this function?" and "What could we achieve by carrying this additional semantic in an IP address?" The questions are also distinct from issues of how the additional semantics can be encoded within an IP

address. All of those issues are, of course, important considerations in the debate about semantic routing, but they form part of the essential groundwork of research into semantic routing itself.

The document "Challenges for the Internet Routing Infrastructure Introduced by Changes in Semantic Routing" [I-D.king-irtf-challenges-in-routing] sets out the challenges for the routing system, and how it might be impacted by the use of semantic routing.

7. Security Considerations

This document is a survey of existing work and so introduces no security considerations of itself. However, many of the proposals referenced either are intended to improve security or have their own security implications. For example:

* In-network path selection, the criteria used for selecting the best path may include security considerations.

* Semantic routing, and applied to specific addresses, may be established using security criteria.

* Physical security may be provided for a site or limited domain simply by not permitting traffic to enter or leave the site. This may be expanded by connecting multiple sites together using tunnels across the Internet to form a Virtual Private Network (VPN) such that the same level of security is shared by all nodes that participate in the VPN provided that the tunnels are themselves secure.

* There are also additional complexities for security when any form of multicast or anycast is used because of issues of address assignment and the formation of security associations.

8. IANA Considerations

This document makes no requests for IANA action.

9. Acknowledgements

Thanks to Stewart Bryant for useful conversations. Luigi Iannone, Robert Raszuk, Ron Bonica, Marie-Jose; Montpetit, Yizhou Li, Toerless Eckert, Tony Li, Joel Halpern, and Carsten Bormann made helpful suggestions. The text on Dyncast is based on suggestions from Dirk Trossen, Luigi Iannone, and Yizhou Li. Toerless Eckert suggested text for the multicast sections.

This work is partially supported by the European Commission under Horizon 2020 grant agreement number 101015857 Secured autonomic traffic management for a Tera of SDN flows (Teraflow).

10. Contributors

Joanna Dang
Email: dangjuanna@huawei.com

Dirk Trossen
Email: dirk.trossen@huawei.com

11. Informative References

[BLIND-FORWARDINGref]
Simsek, I., "On-Demand Blind Packet Forwarding", Paper 30th International Telecommunication Networks and Applications Conference (ITNAC), 2020, 2020, <https://www.computer.org/csdl/proceedings-article/itnac/2020/09315187/1qmfFPPggrC>.

[CLEANSLATEref]
Feldmann, A., "Internet Clean-Slate Design: What and Why?", Paper Annals of telecommunications-annales des telecommunications;64(5):271-6, 2009, 2009, <http://ccr.sigcomm.org/online/files/p59-feldmannA.pdf>.

[CLNPref] "Protocol for providing the connectionless-mode network service: Protocol specification - Part 1", standard ISO/IEC 8473-1:1998, 1998, <https://www.iso.org/standard/30931.html>.

[CONTENT-RTG-MOBILEref]
Liu, H. and W. He, "Rich Semantic Content-oriented Routing for mobile Ad Hoc Networks", Paper The International Conference on Information Networking (ICOIN2014), 2014, 2014, <https://ieeexplore.ieee.org/document/6799682>.

[CONTENTref]
Choi, J., Han, J., and E. Cho, "A survey on content-oriented networking for efficient content delivery", Paper IEEE Communications Magazine, 49(3): 121-127, May 2011., 2011, <https://ieeexplore.ieee.org/iel5/35/5723785/05723809.pdf>.

[EIBPref] Shenoy, N., "Can We Improve Internet Performance? An Expedited Internet Bypass Protocol", Presentation 28th IEEE International Conference on Network Protocols, 2020, <https://icnp20.cs.ucr.edu/Slides/NIPAA/D-3_invited.pptx>.

[GEO-IPref] Dasu, T., Kanza, Y., and D. Srivastava, "Geotagging IP Packets for Location-Aware Software-Defined Networking in the Presence of Virtual Network Functions", Paper 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM SIGSPATIAL 2017), 2017, <https://about.att.com/ecms/dam/sites/labs_research/content/publications/AI_Geotagging_IP_Packets_for_Location.pdf>.

[HICNref] Carofiglio, G., Muscariello, L., Auge, J., Papalini, M., Sardara, M., and A. Compagno, "Enabling ICN in the Internet Protocol: Analysis and Evaluation of the Hybrid-ICN Architecture", Paper Proceedings of the 6th ACM Conference on Information-Centric Networking, 2019., 2019, <https://www.researchgate.net/publication/336344520_Enabling_ICN_in_the_Internet_Protocol_Analysis_and_Evaluation_of_the_Hybrid-ICN_Architecture>.

[I-D.bonica-6man-ext-hdr-update] Bonica, R. and T. Jinmei, "Inserting, Processing And Deleting IPv6 Extension Headers", Work in Progress, Internet-Draft, draft-bonica-6man-ext-hdr-update-07, 24 February 2022, <https://www.ietf.org/archive/id/draft-bonica-6man-ext-hdr-update-07.txt>.

[I-D.boucadair-irtf-sdn-and-semantic-routing] Boucadair, M., Trossen, D., and A. Farrel, "Considerations for the use of SDN in Semantic Routing Networks", Work in Progress, Internet-Draft, draft-boucadair-irtf-sdn-and-semantic-routing-00, 2 February 2022, <https://www.ietf.org/archive/id/draft-boucadair-irtf-sdn-and-semantic-routing-00.txt>.

[I-D.bryant-arch-fwd-layer-ps] Bryant, S., Chunduri, U., Eckert, T., and A. Clemm, "Forwarding Layer Problem Statement", Work in Progress, Internet-Draft, draft-bryant-arch-fwd-layer-ps-04, 24 January 2022, <https://www.ietf.org/archive/id/draft-bryant-arch-fwd-layer-ps-04.txt>.

[I-D.chunduri-lsr-isis-preferred-path-routing]
Chunduri, U., Li, R., White, R., Contreras, L. M., Tantsura, J., and Y. Qu, "Preferred Path Routing (PPR) in IS-IS", Work in Progress, Internet-Draft, draft-chunduri-lsr-isis-preferred-path-routing-07, 12 November 2021, <https://www.ietf.org/archive/id/draft-chunduri-lsr-isis-preferred-path-routing-07.txt>.

[I-D.farrel-irtf-introduction-to-semantic-routing]
Farrel, A. and D. King, "An Introduction to Semantic Routing", Work in Progress, Internet-Draft, draft-farrel-irtf-introduction-to-semantic-routing-04, 25 April 2022, <https://www.ietf.org/archive/id/draft-farrel-irtf-introduction-to-semantic-routing-04.txt>.

[I-D.galis-irtf-sarnet21-report]
Galis, A. and D. Lou, "Semantic Addressing and Routing for Future Networks (SARNET-21) Workshop Report", Work in Progress, Internet-Draft, draft-galis-irtf-sarnet21-report-01, 26 July 2021, <https://www.ietf.org/archive/id/draft-galis-irtf-sarnet21-report-01.txt>.

[I-D.ietf-lisp-rfc6830bis]
Farinacci, D., Fuller, V., Meyer, D., Lewis, D., and A. Cabellos, "The Locator/ID Separation Protocol (LISP)", Work in Progress, Internet-Draft, draft-ietf-lisp-rfc6830bis-38, 7 May 2022, <https://www.ietf.org/archive/id/draft-ietf-lisp-rfc6830bis-38.txt>.

[I-D.ietf-lisp-rfc6833bis]
Farinacci, D., Maino, F., Fuller, V., and A. Cabellos, "Locator/ID Separation Protocol (LISP) Control-Plane", Work in Progress, Internet-Draft, draft-ietf-lisp-rfc6833bis-31, 2 May 2022, <https://www.ietf.org/archive/id/draft-ietf-lisp-rfc6833bis-31.txt>.

[I-D.ietf-v6ops-ipv6-ehs-packet-drops]
Gont, F., Hilliard, N., Doering, G., Kumari, W., Huston, G., and W. (. Liu, "Operational Implications of IPv6 Packets with Extension Headers", Work in Progress, Internet-Draft, draft-ietf-v6ops-ipv6-ehs-packet-drops-08, 11 June 2021, <https://www.ietf.org/archive/id/draft-ietf-v6ops-ipv6-ehs-packet-drops-08.txt>.

[I-D.irtf-panrg-path-properties]
Enghardt, T. and C. Krähenbühl, "A Vocabulary of Path Properties", Work in Progress, Internet-Draft, draft-irtf-panrg-path-properties-05, 7 March 2022, <https://www.ietf.org/archive/id/draft-irtf-panrg-path-properties-05.txt>.

[I-D.irtf-panrg-what-not-to-do]
Dawkins, S., "Path Aware Networking: Obstacles to Deployment (A Bestiary of Roads Not Taken)", Work in Progress, Internet-Draft, draft-irtf-panrg-what-not-to-do-19, 26 March 2021, <https://www.ietf.org/archive/id/draft-irtf-panrg-what-not-to-do-19.txt>.

[I-D.king-irtf-challenges-in-routing]
King, D., Farrel, A., and C. Jacquenet, "Challenges for the Internet Routing Systems Introduced by Semantic Routing", Work in Progress, Internet-Draft, draft-king-irtf-challenges-in-routing-08, 25 April 2022, <https://www.ietf.org/archive/id/draft-king-irtf-challenges-in-routing-08.txt>.

[I-D.kw-rtgwg-satellite-rtg-add-challanges]
King, D. and N. Wang, "Routing and Addressing Challenges Introduced by New Satellite Constellations", Work in Progress, Internet-Draft, draft-kw-rtgwg-satellite-rtg-add-challanges-00, 7 March 2022, <https://www.ietf.org/archive/id/draft-kw-rtgwg-satellite-rtg-add-challanges-00.txt>.

[I-D.lhan-satellite-semantic-addressing]
Han, L., Li, R., Retana, A., Chen, M., and N. Wang, "Satellite Semantic Addressing for Satellite Constellation", Work in Progress, Internet-Draft, draft-lhan-satellite-semantic-addressing-01, 6 March 2022, <https://www.ietf.org/archive/id/draft-lhan-satellite-semantic-addressing-01.txt>.

[I-D.li-dyncast-architecture]
Li, Y., Iannone, L., Trossen, D., Liu, P., and C. Li, "Dynamic-Anycast Architecture", Work in Progress, Internet-Draft, draft-li-dyncast-architecture-03, 7 March 2022, <https://www.ietf.org/archive/id/draft-li-dyncast-architecture-03.txt>.

[I-D.liu-dyncast-ps-usecases]
Liu, P., Eardley, P., Trossen, D., Boucadair, M., Contreras, L. M., and C. Li, "Dynamic-Anycast (Dyncast)

Use Cases and Problem Statement", Work in Progress, Internet-Draft, draft-liu-dyncast-ps-usecases-03, 7 March 2022, <https://www.ietf.org/archive/id/draft-liu-dyncast-ps-usecases-03.txt>.

[I-D.muscariello-intarea-hicn]
Muscariello, L., Carofiglio, G., Augé, J., Papalini, M., and M. Sardara, "Hybrid Information-Centric Networking", Work in Progress, Internet-Draft, draft-muscariello-intarea-hicn-04, 20 May 2020, <https://www.ietf.org/archive/id/draft-muscariello-intarea-hicn-04.txt>.

[I-D.sarathchandra-coin-appcentres]
Trossen, D., Sarathchandra, C., and M. Boniface, "In-Network Computing for App-Centric Micro-Services", Work in Progress, Internet-Draft, draft-sarathchandra-coin-appcentres-04, 26 January 2021, <https://www.ietf.org/archive/id/draft-sarathchandra-coin-appcentres-04.txt>.

[ICNref] Barbera, D., Xylomenos, G., Ververidis, C., Siris, V., and N. Fotiou, "A Survey of information-centric networking research", Paper IEEE Communications Surveys and Tutorials, vol. 16, Iss. 2, Q2 2014, 2014, <https://www.scopus.com/record/display.uri?eid=2-s2.0-84901242669>.

[ISISref] "Intermediate System to Intermediate System intra-domain routeing information exchange protocol for use in conjunction with the protocol for providing the connectionless-mode network service", standard ISO/IEC 10589, 2002, <https://standards.iso.org/ittf/PubliclyAvailableStandards/c030932_ISO_IEC_10589_2002(E).zip>.

[ITUNET2030ref]
"Network 2030 Architecture Framework", Technical Specification ITU-T Focus Group on Technologies for Network 2030, 2020, <https://www.itu.int/en/ITU-T/focusgroups/net2030/Documents/Network_2030_Architecture-framework.pdf>.

[MBONEref] Savetz, K., Randall, N., and Y. Lepage, "MBONE: Multicasting Tomorrow’s Internet", Book IDG, 1996, <https://www.savetz.com/mbone/>.

[MULTICAST-SRref]
Jia, W. and W. He, "A Scalable Multicast Source Routing Architecture for Data Center Networks", Paper IEEE Journal on Selected Areas in Communications, vol. 32, no. 1, pp. 116-123, January 2014, 2014, <https://ieeexplore.ieee.org/document/6799682>.

[NDNref] Zhang, L., Afanasyev, A., and J. Burke, "Named Data Networking", Paper ACM SIGCOMM Computer Communication, Review 44(3): 66-73, 2014, 2014.

[OPENSRNref]
Ren, P., Wang, X., Zhao, B., Wu, C., and H. Sun, "OpenSRN: A Software-defined Semantic Routing Network Architecture", Paper IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Hong Kong, 2015, 2015, <https://www.researchgate.net/publication/308827498_OpenSRN_A_software-defined_semantic_routing_network_architecture>.

[PANRGref] "Path Aware Networking Research Group", RG Path Aware Networking Research Group, <https://datatracker.ietf.org/rg/panrg/about>.

[RESEARCHFIAref]
Pan, J., Paul, S., and R. Jain, "A Survey of the Research on Future Internet Architectures", Paper IEEE Communications Magazine, vol. 49, no. 7, July 2011, 2014, <https://ieeexplore.ieee.org/document/5936152>.

[RFC1069] Callon, R. and H. Braun, "Guidelines for the use of Internet-IP addresses in the ISO Connectionless-Mode Network Protocol", RFC 1069, DOI 10.17487/RFC1069, February 1989, <https://www.rfc-editor.org/info/rfc1069>.

[RFC1112] Deering, S., "Host extensions for IP multicasting", STD 5, RFC 1112, DOI 10.17487/RFC1112, August 1989, <https://www.rfc-editor.org/info/rfc1112>.

[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>.

[RFC2474] Nichols, K., Blake, S., Baker, F., and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, DOI 10.17487/RFC2474, December 1998, <https://www.rfc-editor.org/info/rfc2474>.

[RFC2730] Hanna, S., Patel, B., and M. Shah, "Multicast Address Dynamic Client Allocation Protocol (MADCAP)", RFC 2730, DOI 10.17487/RFC2730, December 1999, <https://www.rfc-editor.org/info/rfc2730>.

[RFC2776] Handley, M., Thaler, D., and R. Kermode, "Multicast-Scope Zone Announcement Protocol (MZAP)", RFC 2776, DOI 10.17487/RFC2776, February 2000, <https://www.rfc-editor.org/info/rfc2776>.

[RFC2909] Radoslavov, P., Estrin, D., Govindan, R., Handley, M., Kumar, S., and D. Thaler, "The Multicast Address-Set Claim (MASC) Protocol", RFC 2909, DOI 10.17487/RFC2909, September 2000, <https://www.rfc-editor.org/info/rfc2909>.

[RFC3618] Fenner, B., Ed. and D. Meyer, Ed., "Multicast Source Discovery Protocol (MSDP)", RFC 3618, DOI 10.17487/RFC3618, October 2003, <https://www.rfc-editor.org/info/rfc3618>.

[RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, DOI 10.17487/RFC4291, February 2006, <https://www.rfc-editor.org/info/rfc4291>.

[RFC4607] Holbrook, H. and B. Cain, "Source-Specific Multicast for IP", RFC 4607, DOI 10.17487/RFC4607, August 2006, <https://www.rfc-editor.org/info/rfc4607>.

[RFC4655] Farrel, A., Vasseur, J.-P., and J. Ash, "A Path Computation Element (PCE)-Based Architecture", RFC 4655, DOI 10.17487/RFC4655, August 2006, <https://www.rfc-editor.org/info/rfc4655>.

[RFC5440] Vasseur, JP., Ed. and JL. Le Roux, Ed., "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, DOI 10.17487/RFC5440, March 2009, <https://www.rfc-editor.org/info/rfc5440>.

[RFC6282] Hui, J., Ed. and P. Thubert, "Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks", RFC 6282, DOI 10.17487/RFC6282, September 2011, <https://www.rfc-editor.org/info/rfc6282>.

[RFC6308] Savola, P., "Overview of the Internet Multicast Addressing Architecture", RFC 6308, DOI 10.17487/RFC6308, June 2011, <https://www.rfc-editor.org/info/rfc6308>.

[RFC6740] Atkinson, RJ. and SN. Bhatti, "Identifier-Locator Network Protocol (ILNP) Architectural Description", RFC 6740, DOI 10.17487/RFC6740, November 2012, <https://www.rfc-editor.org/info/rfc6740>.

[RFC6830] Farinacci, D., Fuller, V., Meyer, D., and D. Lewis, "The Locator/ID Separation Protocol (LISP)", RFC 6830, DOI 10.17487/RFC6830, January 2013, <https://www.rfc-editor.org/info/rfc6830>.

[RFC7094] McPherson, D., Oran, D., Thaler, D., and E. Osterweil, "Architectural Considerations of IP Anycast", RFC 7094, DOI 10.17487/RFC7094, January 2014, <https://www.rfc-editor.org/info/rfc7094>.

[RFC7285] Alimi, R., Ed., Penno, R., Ed., Yang, Y., Ed., Kiesel, S., Previdi, S., Roome, W., Shalunov, S., and R. Woundy, "Application-Layer Traffic Optimization (ALTO) Protocol", RFC 7285, DOI 10.17487/RFC7285, September 2014, <https://www.rfc-editor.org/info/rfc7285>.

[RFC7450] Bumgardner, G., "Automatic Multicast Tunneling", RFC 7450, DOI 10.17487/RFC7450, February 2015, <https://www.rfc-editor.org/info/rfc7450>.

[RFC7665] Halpern, J., Ed. and C. Pignataro, Ed., "Service Function Chaining (SFC) Architecture", RFC 7665, DOI 10.17487/RFC7665, October 2015, <https://www.rfc-editor.org/info/rfc7665>.

[RFC7872] Gont, F., Linkova, J., Chown, T., and W. Liu, "Observations on the Dropping of Packets with IPv6 Extension Headers in the Real World", RFC 7872, DOI 10.17487/RFC7872, June 2016, <https://www.rfc-editor.org/info/rfc7872>.

[RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <https://www.rfc-editor.org/info/rfc8200>.

[RFC8279] Wijnands, IJ., Ed., Rosen, E., Ed., Dolganow, A., Przygienda, T., and S. Aldrin, "Multicast Using Bit Index Explicit Replication (BIER)", RFC 8279, DOI 10.17487/RFC8279, November 2017, <https://www.rfc-editor.org/info/rfc8279>.

[RFC8296] Wijnands, IJ., Ed., Rosen, E., Ed., Dolganow, A., Tantsura, J., Aldrin, S., and I. Meilik, "Encapsulation for Bit Index Explicit Replication (BIER) in MPLS and Non-MPLS Networks", RFC 8296, DOI 10.17487/RFC8296, January 2018, <https://www.rfc-editor.org/info/rfc8296>.

[RFC8300] Quinn, P., Ed., Elzur, U., Ed., and C. Pignataro, Ed., "Network Service Header (NSH)", RFC 8300, DOI 10.17487/RFC8300, January 2018, <https://www.rfc-editor.org/info/rfc8300>.

[RFC8402] Filsfils, C., Ed., Previdi, S., Ed., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", RFC 8402, DOI 10.17487/RFC8402, July 2018, <https://www.rfc-editor.org/info/rfc8402>.

[RFC8596] Malis, A., Bryant, S., Halpern, J., and W. Henderickx, "MPLS Transport Encapsulation for the Service Function Chaining (SFC) Network Service Header (NSH)", RFC 8596, DOI 10.17487/RFC8596, June 2019, <https://www.rfc-editor.org/info/rfc8596>.

[RFC8684] Ford, A., Raiciu, C., Handley, M., Bonaventure, O., and C. Paasch, "TCP Extensions for Multipath Operation with Multiple Addresses", RFC 8684, DOI 10.17487/RFC8684, March 2020, <https://www.rfc-editor.org/info/rfc8684>.

[RFC8736] Venaas, S. and A. Retana, "PIM Message Type Space Extension and Reserved Bits", RFC 8736, DOI 10.17487/RFC8736, February 2020, <https://www.rfc-editor.org/info/rfc8736>.

[RFC8754] Filsfils, C., Ed., Dukes, D., Ed., Previdi, S., Leddy, J., Matsushima, S., and D. Voyer, "IPv6 Segment Routing Header (SRH)", RFC 8754, DOI 10.17487/RFC8754, March 2020, <https://www.rfc-editor.org/info/rfc8754>.

[RFC8763] Rahman, A., Trossen, D., Kutscher, D., and R. Ravindran, "Deployment Considerations for Information-Centric Networking (ICN)", RFC 8763, DOI 10.17487/RFC8763, April 2020, <https://www.rfc-editor.org/info/rfc8763>.

[RFC8799] Carpenter, B. and B. Liu, "Limited Domains and Internet Protocols", RFC 8799, DOI 10.17487/RFC8799, July 2020, <https://www.rfc-editor.org/info/rfc8799>.

[RFC8986] Filsfils, C., Ed., Camarillo, P., Ed., Leddy, J., Voyer, D., Matsushima, S., and Z. Li, "Segment Routing over IPv6 (SRv6) Network Programming", RFC 8986, DOI 10.17487/RFC8986, February 2021, <https://www.rfc-editor.org/info/rfc8986>.

[RINAref] Day, J., "Patterns in Network Architecture: A Return to Fundamentals", Book Prentice Hall, 2008.

[SCIONref] Barbera, D., Chaut, L., Perrig, A., Reischuk, R., and P. Szalachowski, "Patterns in Network Architecture: A Return to Fundamentals", Paper The ACM, vol. 60, no. 6, June 2017, 2017, <https://icnp20.cs.ucr.edu/Slides/NIPAA/D-3_invited.pptx>.

[SEMANTICRTG] Strassner, J., Sung-Su, K., and J. Won-Ki, "Semantic Routing for Improved Network Management in the Future Internet", Book Chapter Springer, Recent Trends in Wireless and Mobile Networks, 2010, 2010, <https://link.springer.com/chapter/10.1007/978-3-642-14171-3_14>.

[TERASTREAMref] Zaluski, B., Rajtar, B., Habjani, H., Baranek, M., Slibar, N., Petracic, R., and T. Sukser, "Terastream implementation of all IP new architecture", Paper 36th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2013, 2013, <https://ieeexplore.ieee.org/document/6596297>.

Authors' Addresses

Daniel King
Lancaster University
United Kingdom
Email: d.king@lancaster.ac.uk

Adrian Farrel
Old Dog Consulting
United Kingdom
Email: adrian@olddog.co.uk

Network Working Group
Internet-Draft
Intended status: Informational
Expires: 7 July 2024

L. Han, Ed.
R. Li
A. Retana
Futurewei Technologies, Inc.
M. Chen
L. Su
Tianji. Jiang
China Mobile

4 January 2024

Problems and Requirements of Satellite Constellation for Internet
draft-lhan-problems-requirements-satellite-net-06

Abstract

This document presents the detailed analysis about the problems and requirements of satellite constellation used for Internet. It starts from the satellite orbit basics, coverage calculation, then it estimates the time constraints for the communications between satellite and ground-station, also between satellites. How to use satellite constellation for Internet is discussed in detail including the satellite relay and satellite networking. The problems and requirements of using traditional network technology for satellite network integrating with Internet are finally outlined.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

Satellite constellation for Internet is emerging. Even there is no constellation network established completely yet at the time of the publishing of the draft (June 2021), some basic internet service has been provided and has demonstrated competitive quality to traditional broadband service.

This memo will analyze the challenges for satellite network used in Internet by traditional routing and switching technologies. It is based on the analysis of the dynamic characters of both ground-station-to-satellite and inter-satellite communications and its impact to satellite constellation networking.

The memo also provides visions for the future solution, such as in routing and forwarding.

The memo focuses on the topics about how the satellite network can work with Internet. It does not focus on physical layer technologies (wireless, spectrum, laser, mobility, etc.) for satellite communication.

2. Terminology

LEO	Low Earth Orbit with the altitude from 180 km to 2000 km.
VLEO	Very Low Earth Orbit with the altitude below 450 km
MEO	Medium Earth Orbit with the altitude from 2000 km to 35786 km
GEO	Geosynchronous orbit with the altitude 35786 km
GSO	Geosynchronous satellite on GEO
ISL	Inter Satellite Link
ISLL	Inter Satellite Laser Link

3GPP	3rd Generation Partner-ship Project
NTN	Non-Terrestrial Network, it includes satellite networks (satellite could be on GEO, MEO, LEO or VLEO), high altitude platform systems (HAPS) and other types of air-to-ground networks
EIRP	Effective isotropic radiated power
P2MP	Point to Multiple Points
GS	Ground Station, a device on ground connecting the satellite. In the document, GS will hypothetically provide L2 and/or L3 functionality in addition to process/send/receive radio wave. It might be different as the reality that the device to process/send/receive radio wave and the device to provide L2 and/or L3 functionality could be separated.
SGS	Source ground station. For a specified flow, a ground station that will send data to a satellite through its uplink.
DGS	Destination ground station. For a specified flow, a ground station that is connected to a local network or Internet, it will receive data from a satellite through its downlink and then forward to a local network or Internet.
PGW	Packet Gateway
UPF	User Packet Function
NodeB	The base station in 3G
eNodeB	The base station in 4G
gNB	gNodeB, the base station in 5G
PE router	Provider Edge router
CE router	Customer Edge router
P router	Provider router
LSA	Link-state advertisement

LSP	Link-State PDUs
L1	Layer 1, or Physical Layer in OSI model [OSI-Model]
L2	Layer 2, or Data Link Layer in OSI model [OSI-Model]
L3	Layer 3, or Network Layer in OSI model [OSI-Model], it is also called IP layer in TCP/IP model
BGP	Border Gateway Protocol [RFC4271]
eBGP	External Border Gateway Protocol, two BGP peers have different Autonomous Number
iBGP	Internal Border Gateway Protocol, two BGP peers have same Autonomous Number
IGP	Interior gateway protocol, examples of IGPs include Open Shortest Path First (OSPF [RFC2328]), Routing Information Protocol (RIP [RFC2453]), Intermediate System to Intermediate System (IS-IS [RFC7142]) and Enhanced Interior Gateway Routing Protocol (EIGRP [RFC7868]).

3. Overview

The traditional satellite communication system is composed of few GSO and ground stations. For this system, each GSO can cover 42% Earth's surface [GEO-Coverage], so as few as three GSO can provide the global coverage theoretically. With so huge coverage, GSO only needs to amplify signals received from uplink of one ground station and relay to the downlink of another ground station. There is no inter-satellite communications needed. Also, since the GSO is stationary to the ground station, there is no mobility issue involved.

Recently, more and more LEO and VLEO satellites have been launched, they attract attentions due to their advantages over GSO and MEO in terms of higher bandwidth, lower cost in satellite, launching, ground station, etc. Some organizations [ITU-6G][Surrey-6G][Nttdocomo-6G] have proposed the non-terrestrial network using LEO, VLEO as important parts for 6G to extend the coverage of Internet. 3GPP has been working on the NTN integration with 5G and beyond. SpaceX has started to build the satellite constellation called StarLink that will deploy over 10 thousand LEO and VLEO satellites finally [StarLink]. China also started to request the spectrum from ITU to establish a constellation that has 12992 satellites [China-constellation]. European Space Agency (ESA) has proposed "Fiber in the sky" initiative to connect satellites with fiber network on Earth [ESA-HydRON].

When satellites on MEO, LEO and VLEO are deployed, the communication problem becomes more complicated than for GSO satellites. This is because the altitude of MEO/LEO/VLEO satellites are much lower. As a result, the coverage of each satellite is much smaller than for GSO, and the satellite is moving very fast on the ground reference and not relatively stationary to the ground. This will lead to:

1. More satellites than GSO are needed to provide the global coverage. Appendix A will brief the satellite orbit parameters; analyze the coverage area, and the minimum number of satellites required to cover the earth surface; discuss the real deployment for LEO satellite network.

2. The point-to-point communication between satellite and ground station can only last a few minutes. Mobility issue has to be considered. Detailed analysis about the lifetime of communication is done in Appendix B.1.

3. The inter-satellite communication is needed, and all satellites need to form a network. details are described in Appendix B.2 that includes the communication between satellites on different orbit and different geographic areas.

In Section 4, we will discuss couple of topics of satellite network integration with Internet, such as using satellite network for broadband access and wireless access, the current 3GPP works for satellite network in 5G and beyond.

Finally, the problems and requirements for satellite network integration with Internet will be discussed and analyzed in Section 5.

As the 1st satellite constellation company in history, the SpaceX/ StarLink will be inevitably mentioned in the draft. But it must be noted that all information about SpaceX/StarLink in the draft are from the public. Authors of the draft have no relationship or relevant inside knowledge of SpaceX/Starlink.

4. Satellite Network Integrated with Internet

Since there is no complete satellite network established yet, all following analysis is based on the predictions from the traditional GEO communication. The analysis also learnt how other type of network has been used in Internet, such as Broadband access network, Mobile access network, Enterprise network and Service Provider network.

To integrate the satellite network with Internet, many other technologies are needed to provide the functions on different layeres. Currently, there are four major international SDO (standard organizations) involved in the development of different technologies: IETF, 3GPP, CCSDS (Consultative Committee for Space Data Systems [CCSDS]), DVB (Digital Video Broadcast [DVB]). Section 4.1 will discuss the different protocol stacks based on different combinations of technologies from different SDOs.

As a criteria to be part of Internet, any device connected to any satellite should be able to communicate with any public IP4 or IPv6 address in Internet. There could be three types of methods to deliver IP packet from source to destination by satellite:

1. Data packet is relayed between ground station and satellite. For this method, there is no inter-satellite communication and networking. Data packet is bounced once or couple times between ground stations and satellites until the packet arrives at the destination in Internet.

2. Data packet is delivered by inter-satellite networking. For this method, the data packet traverses with multiple satellites connected by ISL and inter-satellite networking is used to deliver the packet to the destination in Internet.

3. Both satellite relay and inter-satellite networking are used. For this method, the data packet is relayed in some segments and traverse with multiple satellites in other segments. It is a combination of the method 1 and method 2.

Using the above methods for IP packet delivery via satellite network, we will have two typical use cases for satellite network. One is for the general broadband access (see Section 4.2), another is for the integration with 3GPP wireless network including 4G and 5G (see Section 4.3 and Section 4.4).

It must be noted that we use 3GPP as an use case example does not mean other technologies cannot be used, e.g., using DVB instead of 3GPP for the satellite network integration. We use 3GPP is because 3GPP has done the most thorough research and produced lots of solutions for satellite networking, such as using 5G NR for satellite up link and down link, use transparent pay load and regenerative pay load for different scenarios, etc. (see Section 4.3).

4.1. Protocol Stack for Satellite Networking with Different Technologies

Figure 1 illustrates three typical protocol statcks that use different technology combinations. This does not include the use of private technologies for wireless and Link Layer such as Starlink.

The stacks show obviously that TCP/IP is the common technologies, IETF has to (at least) provide the L3 and L4 technologies for satellite networking integrated with Internet.

```
  +-------------------+ +--------------------+ +-------------------+
  |5G upper layers(SA)| | CCSDS upper layers | |  DVB upper layers |
  +-------------------+ +--------------------+ +-------------------+
  |      TCP/UDP      | |      TCP/UDP       | |      TCP/UDP      |
  |         IP        | |         IP         | |        IP         |
  +-------------------+ +--------------------+ +-------------------+
  |       SDAP        | |        IPoC        | |      IPoGSE       |
  |       PDCP        | |                    | |                   |
  |        RLC        | |        USLP        | |    DVB MPE/GSE    |
  |        MAC        | |                    | |                   |
  +-------------------+ +--------------------+ +-------------------+
  |      3GPP RF;     | |      CCSDS RF;     | |   DVB S2/S2X/T2;  |
  | 3GPP Spectrum;FSO | | CCSDS Spectrum;FSO | | DVB Spectrum; FSO |
  +-------------------+ +--------------------+ +-------------------+
        3GPP Stacks           CCSDS Statks            DVB Stacks
```

Figure 1: Protocol Statcks for Different Standard Technologies

Some meaning of symbols in Figure 1 are as follows:

SA Service Architecture

SDAP SService Data Adaption Protocol

PDCP	Packet Data Convergence Protocol
RLC	Radio Link Control
MAC	Medium Access Control
RF	Radio Frequency
FSO	Free Space Optics, provided by ITU
IPoC	IP over CCSDS
USLP	Unified Space Link Protocol
IPoGSE	IP over Generic Stream Encapsulation
MPE	Multiprotocol Encapsulation
GSE	Generic Stream Encapsulation
S2	Digital Video Broadcasting Satellite Second Generation
S2X	Digital Video Broadcasting Satellite Second Generation Extension
T2	Digital Video Broadcasting Satellite Second Generation Terrestrial

4.2. Use Satellite Network for Broadband Access

For this use case, the end user terminal or local network is connected to a ground station, and another ground station is connected to Internet. Two ground stations will have IP connectivity via a satellite network. The satellite network could be by satellite relays or by inter-satellite network.

Follows are typical deployment scenarios that a Satellite network is used for broadband access of Internet.

1. The end user terminal access Internet through satellite relay (Figure 2 for one satellite relay, Figure 3 for multiple satellite relay).

2. The end user terminal access Internet through inter-satellite-networking (Figure 4).

3. The local network access Internet through satellite relay (Figure 5 for one satellite relay, Figure 6 for multiple satellite relay).

4. The local network access Internet through inter-satellite-networking (Figure 7).

```
                    S1----\            /-----------\
                   /       \          /             \
            T---GW--GS1--S2--GS2-------PE   Internet   +
                   \       /          \             /
                    \---S3/            \-----------/
```

Figure 2: End user terminal access Internet through one satellite relay

```
                  S1----\    S4----\       /-----------\
                 /       \  /       \     /             \
          T---GW--GS1--S2--GS2---S5--GS3---PE   Internet   +
                 \       /  \       /     \             /
                  \---S3/    \---S6/       \-----------/
```

Figure 3: End user terminal access Internet through multiple satellite relay

```
             S1-----S2-----S3--\            /----------\
            /                  \          /            \
      T---GW--GS1--S4----S5---S6---GS2-------PE  Internet   +
            \                  /          \            /
             \---S7----S8----S9/            \----------/
```

Figure 4: End user terminal access Internet through inter-satellite-networking

```
       /-----------\           S1----\           /-------\
      /             \         /       \         /         \
     + Local network CE------GS1--S4--GS2-------PE Internet +
      \             /         \       /         \         /
       \-----------/           \---S7/           \-------/
```

Figure 5: Local network access Internet through one satellite relay

```
       /-----------\         S1----\   S4----\       /-------\
      /             \       /       \  /      \     /         \
     + Local network CE----GS1--S2--GS2--S5--GS3---PE Internet +
      \             /       \       / \       /     \         /
       \-----------/         \---S3/   \---S6/       \-------/
```

Figure 6: Local network access Internet through multiple satellite relay

```
   /-----------\          S1-----S2-----S3---\            /------\
  /             \         /                   \          /        \
 + Local network CE------GS1--S4----S5---S6---GS2-------PE Internet+
  \             /         \                   /          \        /
   \-----------/           \---S7----S8----S9/            \------/
```

Figure 7: Local network access Internet through inter-satellite-networking

In above Figure 2 to Figure 7, the meaning of symbols are as follows:

T	The end user terminal
GW	Gateway router
GS1, GS2, GS3	Ground station with L2/L3 routing/switch functionality.
S1 to S9	Satellites
PE	Provider Edge Router
CE	Customer Edge Router

The above configuration may have different variations, e.g., the GW and GS functions can be merged into one ohysical devices.

4.3. Use Satellite Network with 3GPP Wireless Access Network

For this use case, the wireless access network (4G, 5G) defined in 3GPP is used with satellite network. By such integration, a user terminal or local network can access Internet via 3GPP wireless network and satellite network. The End user terminal or local network access Internet through satellite network and Mobile Access Network. There are two cases: 1) From mobile access network to satellite network or 2) From satellite network to mobile access network, Satellite network includes inter satellite network and relay network. See Figure 8 for mobile access network to satellite network, and Figure 9 for satellite network to mobile access network.

```
+--------------+    +-------------+    +---------+    +--------+
|    T or      |    |Mobile Access|    |Satellite|    |Internet|
| Local network+----+  Network    +----+ Network +----+        |
+--------------+    +-------------+    +---------+    +--------+
```

Figure 8: End user terminal or local network access Internet through Mobile Access Network and Satellite Network

```
+--------------+   +---------+   +-------------+   +--------+
|    T or      |   |Satellite|   |Mobile Access|   |Internet|
| Local network+---+ Network +---+  Network    +---+        |
+--------------+   +---------+   +-------------+   +--------+
```

Figure 9: End user terminal or local network access Internet through Satellite Network and Mobile Access Network

4.4. Recent Development and Study in 3GPP for Satellite Network

3GPP SA Working Groups (WG) feature a couple of satellite-related projects (or SIDs).

For Release 18, 3GPP has finished the project 'Study on 5G System with Satellite Backhaul' [TR-23.700-27] and 'Study on 5GC enhancement for satellite access Phase 2' [TR-23.700-28].

For Release 19, 3GPP will study more topics for satellite network used for 5G system, such as Regenerative payload generic architecture study, Store and Forward Satellite operation, etc.

One key aspect is to investigate the potential architecture requirements and enhancements to deploy UPFs on satellites (LEO/MEO/GEO) with gNBs on the ground. Specifically, it targets at enhancing the local-switching capability for UE-to-UE data communication when UEs are served by UPFs on-board satellite(s). Similarly, the SA1 WG proposed a new satellite-based SID in which the service end points (could also be called UEs in a broader sense) may continuously move in a fast way. The UEs can be ships, boats, and cars, etc., which are located in remote regions that need the connection to LEO's for achieving communication.

In all the SIDs, satellite based backhaul is important for mission critical scenarios in remote areas. Here, we want to clarify that while 3GPP documents TS 23.501 [TS-23.501] and 23.502 [TS-23.502] specify that a ground base station, i.e., gNB, may have multiple types of satellite backhauls (BH), e.g., GEO BH, LEO BH and LEO-BH with ISL, this use case focuses specifically on the LEO-BH with ISL. ISL stands for inter-satellite link.

Clearly, when a satellite backhaul involves multi-hop ISL path connected via different satellites, the capabilities provided by the satellite path would be changed and adjusted dynamically. For example, in the LEO case, the peering relationship between two neighboring satellites changes roughly every 5 minutes thanks to the

orbital movement (see Table 2). This will definitely impair the networking performance and stability, and, in worst case, may cause the loss of connectivity. Even if some overlay tunneling mechanisms could be used to address the multi-hop ISL issue, the extra delay and potentially less bandwidth as introduced naturally by the ever-changing backhaul path would still impact the traffic engineering over the links.

The following diagram Figure 10 demonstrate the dynamic characteristics of satellite backhaul between two UEs. In the figure, UEs are connected, via gNBs, to UPFs on-board satellites. Both UPFs are connected via multi-hop ISLs to the 5G core (5GC) on the ground. There are two different multi-hop ISL paths: o A UE has to rely on a multi-hop ISL path to connect to 5GC on the ground. o When two UEs intend to communicate via the local data switching on satellite(s), some new ISL-based peering has to be established which would bring in the multi-hop ISL scenario. For example, the ISL between the Sat#1 and Sat#2 helps form a multi-hop path (marked N19 in the diagram) between the two UEs. Note that if the UPF-based local data switching involves only one UPF, then it is designated as intra-UPF local switching and relatively simpler. This is compared to the case of inter-UPF local switching as shown in the diagram.

```
     Sat#: Satellite                    GS:  Ground Station
     UPF:  User Plane Function (5G)     gNB: Next Generation NodeB

                         Sat#1
                  +----------+       +--------+
                  |  UPF#1   | (ISL) | Sat#.. |
    UE#1--gNB#1--|(on-board)|- - - -|        |-----+
                  +----------+       +--------+     |
                       :                            |      --------
                       :(N19)                       v     /        \
                       :(ISL)                      GS ---+ 5G Core +
                       :                            ^     \        /
                  +----------+       +--------+     |      --------
                  |  UPF#2   | (ISL) | Sat#.. |     |
    UE#2--gNB#2--|(on-board)|- - - -|        |-----+
                  +----------+       +--------+
                     Sat#2
```

Figure 10: Use Satellite network as back haul for 5G

In this diagram, both UEs are served by different satellite backhauls. If the local data switching via LEO UPFs on-board could be established (via the N19 ISL forwarding), then the system efficiency and QoE improvement would be achieved. Here, since UEs are served by different satellites, a multi-hop ISL scenario must be

supported. But, this scenario posts challenges due to the dynamic satellite network topology and distinguished transmission capabilities from different satellites.

For example, if the UE-to-UE session has to maintain a service over longer time (> 5 minutes) such that the Sat#1 and Sat#2 move apart, then a new ISL path with potentially a new N19-ISL might be established. In worst case, if newly-involved satellites in the path happen to be polar-orbit ones and they do not support cross-seam ISLs, the communication latency may change dramatically when cross-seam transits or leaves. In another example, if both UEs belong to the same entity and need to form a 5G-VN group, then the 5G LAN-type service with PSA UPF-based local-switching must be applied among them.

Regardless, more efficient satellite communication mechanisms must be adopted, e.g., running efficient satellite-based routing protocols, establishing tunnels between LEO UPFs on-board, etc., for better local-data switching.

Further, 5GS may collaborate with satellite networks to improve QoS. One 5GC NF (i.e., SMF) can initiate UP path monitoring, and accordingly receive UP path monitoring results indicating observed delay. After that, the SMF takes corresponding actions like further verifying network statistics, updating sessions, etc. The coordination with the satellite networks would improve the process, which suggests satellites networks respond better to the (monitor-based) polling from 5GS.

One more thing we want to point out is that, while the propagation delay of satellite backhaul paths may change dramatically with the movement of satellite, this kind of change normally be periodic and can be well predicated based on the operation information of satellite constellation. Thus, making use of these information would also help for better services.

5. Problems and Requirements for Satellite Constellation for Internet

As described in Section 4, satellites in a satellite constellation can either relay internet traffic or multiple satellites can form a network to deliver internet traffic. More detailed analysis are in following sub sections. There might have multiple solutions for each method described in Section 4, following contexts only discuss the most plausible solution from networking perspectives.

Section 5.1 will list the common problems and requirements for both satellite relay and satellite networking.

Section 5.2 and Section 5.3 will describe key problems, requirement and potential solution from the networking perspective for these two cases respectively.

5.1. Common Problems and Requirements

For both satellite relay and satellite networking, satellite-ground-station communication must be used, so, the problems and requirements for the satellite-ground-station communication is common and will apply for both methods.

When one satellite is communicating with ground station, the satellite only needs to receive data from uplink of one ground station, process it and then send to the downlink of another ground station. Figure 2 illustrates this case. Normally microwave is used for both links.

Additionally, from the coverage analysis in Appendix A.2 and real deployment in Appendix A.3, we can see one ground station may communicate with multiple satellites. Similarly, one satellite may communicate with multiple ground stations. The characters for satellite-ground-station communication are:

1. Satellite-ground-station communication is P2MP.
   Since microwave physically is the carrier of broadcast communication, one satellite can send data while multiple ground stations can receive it. Similarly, one ground station can send data and multiple satellites can receive it.

2. Satellite-ground-station communication is in open space and not secure.
   Since electromagnetic fields for microwave physically are propagating in open space. The satellite-ground-station communication is also in open space. It is not secure naturally.

3. Satellite-ground-station communication is not steady.
   Since the satellite is moving with high speed, from Appendix B.1, the satellite-ground-station communication can only last a certain period of time. The communication peers will keep changing.

4. Satellite-to-Satellite communication is not steady.
   For some satellites, even they are in the same altitude and move in the same speed, but they move in the opposite direction, from Appendix B.2.2, the satellite-to-satellite communication can only last a certain period of time. The communication peers will keep changing.

5. Satellite-to-Satellite distance is not steady.
   For satellites with the same altitude and same moving direction, even their relative position is steady, but the distance between satellites are not steady. This will lead to the inter-satellite-communication's bandwidth and latency keep changing.

6. Satellite physical resource is limited.
   Due to the weight, complexity and cost constraint, the physical resource on a satellite, such as power supply, memory, link speed, are limited. It cannot be compared with the similar device on ground. The design and technology used should consider these factors and take the appropriate approach if possible.

The requirements of satellite-ground-station communication are:

R1. The bi-directional communication capability
   Both satellites and ground stations have the bi-directional communication capability

R2. The identifier for satellites and ground stations
   Satellites and ground stations should have Ethernet and/or IP address configured for the device and each link. More detailed address configuration can be seen in each solution.

R3. The capability to decide where the IP packet is forwarded to.
   In order to send Internet traffic or IP date to destination correctly, satellites and ground station must have Ethernet hub or switching or IP routing capability. More detailed capability can be seen in each solution.

R4. The protocol to establish the satellite-ground-station communication.
   For security and management purpose, the satellite-ground-station communication is only allowed after both sides agree through a protocol. The protocol should be able to establish a secured channel for the communication when a new communication peer comes up. Each ground station should be able to establish multiple channels to communicate with multiple satellites. Similarly, each satellite should be able to establish multiple channels to communicate to multiple ground stations.

R5. The protocol to discover the state of communication peer.
   The discover protocol is needed to detect the state of communication peer such as peer's identity, the state of the peer and other info of the peer. The protocol must be running securely without leaking the discovered info.

R6. The internet data packet is forwarded securely.
When satellite or ground station is sending the IP packet to its peer, the packet must be relayed securely without leaking the user data.

R7. The internet data packet is processed efficiently on satellite
Due to the resource constraint on a satellite, the packet may need more efficient mechanism to be processed on satellite. The process on satellite should be very minimal and offloaded to ground as much as possible.

5.2. Satellite Relay

One of the reasons to use satellite constellation for internet access is it can provide shorter latency than using the fiber underground. But using ISL for inter-satellite communication is the premise for such benefit in latency. Since the ISL is still not mature and adopted commercially, satellite relay is a only choice currently for satellite constellation used for internet access. In [UCL-Mark-Handley], detailed simulations have demonstrated better latency than fiber network by satellite relay even the ISL is not present.

5.2.1. One Satellite Relay

One satellite relay is the simplest method for satellite constellation to provide Internet service. By this method, IP traffic will be relayed by one satellite to reach the DGS and go to Internet.

The solution option and associated requirements are:

S1. The satellite only does L1 relay or the physical signal process.

For this solution, a satellite only receives physical signal, amplify it and broadcast to ground stations. It has no further process for packet, such as L2 packet compositing and processing, etc. All packet level work is done only at ground station. The requirements for the solution are:

R1-1. SGS and BGS are configured as IP routing node. Routing protocol is running in SGS and BGS
SGS and BGS is a IP peer for a routing protocol (IGP or BGP). SGS will send internet traffic to DGS as next hop through satellite uplink and downlink.

R1-2. DGS must be connected with Internet.
DGS can process received packet from satellite and forward the packet to the destination in Internet.

In addition to the above requirements, following problem should be solved:

P1-1. IP continuity between two ground stations
This problem is that two ground stations are connected by one satellite relay. Since the satellite is moving, the IP continuity between ground stations is interrupted by satellite changing periodically. Even though this is not killing problem from the view point that IP service traditionally is only a best effort service, it will benefit the service if the problem can be solved. Different approaches may exist, such as using hands off protocols, multipath solutions, etc.

S2. The satellite does the L2 relay or L2 packet process.

For this solution, IP packet is passing through individual satellite as an L2 capable device. Unlike in the solution S1, satellite knows which ground station it should send based on packet's destination MAC address after L2 processing. The advantage of this solution over S1 is it can use narrower beam to communicate with DGS and get higher bandwidth and better security. The requirements for the solution are:

R2-1. Satellite must have L2 bridge or switch capability
In order to forward packet to properly, satellite should run some L2 process such as MAC learning, MAC switching. The protocol running on satellite must consider the fast movement of satellite and its impact to protocol convergence, timer configuration, table refreshment, etc.

R2-2. same as R1-1 in S1

R2-3. same as R1-2 in S1

In addition to the above requirements, the problem P1-1 for S1 should also apply.

5.2.2. Multiple Satellite Relay

For this method, packet from SGS will be relayed through multiple intermediate satellites and ground station until reaching a DGS.

This is more complicated than one satellite relay described in Section 5.2.1.

One general solution is to configure both satellites and ground-stations as IP routing nodes, proper routing protocols are running in this network. The routing protocol will dynamically determine forwarding path. The obvious challenge for this solution is that all links between satellite and ground station are not static, according to the analysis in Appendix B.1, the lifetime of each link may last only couple of minutes. This will result in very quick and constant topology changes in both link state and IP adjacency, it will cause the distributed routing algorithms may never converge. So this solution is not feasible.

Another plausible solution is to specify path statically. The path is composed of a serials of intermediate ground stations plus SGS and DGS. This idea will make ground stations static and leave the satellites dynamic. It will reduce the fluctuation of network path, thus provide more steady service. One variant for the solution is whether the intermediate ground stations are connected to Internet. Separated discussion is as below:

S1. Manual configuring routing path and table

For this solution, the intermediate ground stations and DGS are specified and configured manually during the stage of network planning and provisioning. Following requirements apply:

R1-1. Specify a path from SGS to DGS via a list of intermediate ground stations.
The specified DGS must be connected with internet. Other specified intermediate ground stations does not have to

R1-2. All Ground stations are configured as IP routing node.
Static routing table on all ground stations must be pre-configured, the next hop of routes to Internet destination in any ground station is configured to going through uplink of satellite to the next ground station until reaching the DGS.

R1-3. All Satellites are configured as either L1 relay or L2 relay.
The Satellite can be configured as L1 relay or L2 relay described in S1 and S2 respectively in Section 5.2.1

In addition to the above requirements, the problem P1-1 in Section 5.2.1 should also apply.

S2. Automatic decision by routing protocol.

This solution is only feasible after the IP continuity problem (P1-1 in Section 5.2.1) is solved. Following requirements apply:

R2-1. All Ground stations are configured as IP routing node. Proper routing protocols are configured as well.
The satellite link cost is configured to be lower than the ground link. In such a way, the next hop of routes for the IP forwarding to Internet destination in any ground station will be always going through the uplink of satellite to the next ground station until reaching the DGS.

R2-2. All Satellites are configured as either L1 relay or L2 relay.
The Satellite can be configured as L1 relay or L2 relay described in S1 and S2 respectively in Section 5.2.1

In addition to the above requirements, the problem P1-1 in Section 5.2.1 should also apply.

5.3. Satellite Networking by ISL

In the draft, satellite Network is defined as a network that satellites are inter-connected by inter-satellite links (ISL). One of the major difference of satellite network with the other type of network on ground (telephone, fiber, etc.) is its topology and links are not stationary, some new issues have to be considered and solved. Follows are the factors that impact the satellite networking.

5.3.1. L2 or L3 network

The 1st question to answer is should the satellite network be configured as L2 or L3 network? As analyzed in Appendix A.2 and Appendix A.3, since there are couple of hundred or over ten thousand satellites in a network, L2 network is not a good choice, instead, L3 or IP network is more appropriate for such scale of network.

5.3.2. Inter-satellite-Link Lifetime

If we assume the orbit is circular and ignore other trivial factors, the satellite speed is approximately determined by the orbit altitude as described in the Appendix B.1. The satellite orbit can determine if the dynamic position of two satellites is within the range of the inter-satellite communication. That is 2000km for laser communication [Laser-communication-range] by Inter Satellite Laser Link (ISLL).

When two satellites' orbit planes belong to the same group, or two orbit planes share the same altitude and inclination, and when the satellites move in the same direction, the relative positions of two satellites are relatively stationary, and the inter-satellite communication is steady. But when the satellites move in the

opposite direction, the relative positions of two satellites are not stationary, the communication lifetime is couple of minutes. The Appendix B.2.2 has analyzed the scenario.

When two satellites' orbit planes belong to the different group, or two orbit planes have different altitude, the relative position of two satellite are unstable, and the inter-satellite communication is not steady. As described in Appendix B.2, The life of communication for two satellites depends on the following parameters of two satellites:

1. The speed vectors.

2. The altitude difference

3. The intersection angle

From the examples shown in Table 4 to Table 7, we can see that the lifetime of inter-satellite communication for the different group of orbit planes are from couple of hundred seconds to about 18 hours. This fact will impact the routing technologies used for satellite network and will be discussed in Section 5.3.3.

5.3.3. Problems for Traditional Routing Technologies

When the satellite network is integrated with Internet by traditional routing technologies, following provisioning and configuration (see Figure 11) will apply:

1. The ground stations connected to local network and internet are treated as PE router for satellite network (called PE_GS1 and PE_GS2 in the following context), and all satellites are treated as P router.

2. All satellites in the network and ground stations are configured to run IGP.

3. The eBGP is configured between PE_GS and its peered network's PE or CE.

The work on PE_GS1 are:

* The local network routes are received at PE_GS1 from CE by eBGP. The routes are redistributed to IGP and then IGP flood them to all satellites. (Other more efficient methods, such as iBGP or BGP reflectors are hard to be used, since the satellite is moving and there is no easy way to configure a full meshed iBGP session for all satellites, or configure one satellite as BGP reflector in satellite network.)

* The internet routes are redistributed from IGP to eBGP running on PE_GS1, and eBGP will advertise them to CE.

The work on PE_GS2 are:

* The Internet routes are received at PE_GS2 from PE by eBGP. The routes are redistributed to IGP and then IGP flood them to all satellites. (Similar as in PE_GS1, Other more efficient methods, such as iBGP or BGP reflector cannot be used.)

* The local network routes are redistributed from IGP to eBGP running on PE_GS2, and eBGP will advertise them to Internet.

```
  /--------\             S1---S2----S3----\               /------\
 /          \            /    IGP domain   \             /        \
+ Local net CE--eBGP--PE_GS1---S4---S5---PE_GS2--eBGP--PE Internet +
 \          /            \                 /             \        /
  \--------/              \---S6---S7---S8/               \------/
```

Figure 11: Local access Internet through inter-satellite-networking

Local access Internet through inter-satellite-networking

On PE-GS1, due to the fact that IGP link between PE_GS1 and satellite is not steady; this will lead to following routing activity:

1. When one satellite is connecting with PE_GS1, the satellite and PE_GS1 form a IGP adjacency. IGP starts to exchange the link state update.

2. The local network routes received by eBGP in PE_GS1 from CE are redistributed to IGP, and IGP starts to flood link state update to all satellites.

3. Meanwhile, the Internet routes learnt from IGP in PE_GS1 will be redistributed to eBGP. eBGP starts to advertise to CE.

4. Every satellite will update its routing table (RIB) and forwarding table (FIB) after IGP finishes the SPF algorithm.

5. When the satellite is disconnecting with PE-GS1, the IGP adjacency between satellite and PE_GS1 is gone. IGP starts to exchange the link state update.

6. The routes of local network and satellite network that were redistributed to IGP in step 2 will be withdrawn, and IGP starts to flood link state update to all satellites.

7. Meanwhile, the Internet routes previously redistributed to eBGP in step 3 will also be withdrawn. eBGP starts to advertise route withdraw to CE.

8. Every satellite will update its routing table (RIB) and forwarding table (FIB) after the SPF algorithm.

Similarly on PE_GS2, due to the fact that IGP link between PE_GS2 and satellite is not steady; this will lead to following routing activity:

1. When one satellite is connecting with PE_GS2, the satellite and PE_GS2 form a IGP adjacency. IGP starts to exchange the link state update.

2. The Internet routes previously received by eBGP in PE_GS2 from PE are redistributed to IGP, IGP starts to flood the new link state update to all satellites.

3. Meanwhile, the routes of local network and satellite network learnt from IGP in PE_GS2 will be redistributed to eBGP. eBGP starts to advertise to Internet peer PE.

4. Every satellite will update its routing table (RIB) and forwarding table (FIB) after IGP finishes the SPF algorithm.

5. When the satellite is disconnecting with PE-GS2, the IGP adjacency between satellite and PE_GS2 is gone. IGP starts to exchange the link state update.

6. The internet routes previously redistributed to IGP in step 2 will be withdrawn, and IGP starts to flood link state update to all satellites

7. Meanwhile, the routes of local network and satellite network previously redistributed to eBGP in step 3 will also be withdrawn. eBGP starts to advertise route withdraw to PE.

8. Every satellite will update its routing table (RIB) and forwarding table (FIB) after the SPF algorithm.

For the analysis of detailed events above, the estimated time interval between event 1 and 5 for PE_GS1 and PE_GS2 can use the analysis in Appendix B.1. For example, it is about 398s for LEO and 103s for VLEO. Within this time interval, the satellite network including all satellites and two ground stations must finish the works from 1 to 4 for PE_GS1 and PE_GS2. The normal internet IPv6 and IPv4 BGP routes size are about 850k v4 routes + 100K v6 routes [BGP-Table-Size]. There are couple critical problems associated with the events:

P1. Frequent IGP update for its link cost
Even for satellites in different orbit with the steady relative positions, the distance between satellites is keep changing. If the distance is used as the link cost, it means the IGP has to update the link cost frequently. This will make IGP keep running and update its routing table.

P2. Frequent IGP flooding for the internet routes
Whenever the IGP adjacency changes (step 1 and 5 for PE_GS2), it will trigger the massive IGP flooding for the link state update for massive internet routes learnt from eBGP. This will result in the IGP re-convergency, RIB and FIP update.

P3. Frequent BGP advertisement for the internet routes
Whenever the IGP adjacency changes (step 3 and 7 for PE_GS1), it will trigger the massive BGP advertisement for the internet routes learnt from IGP. This will result in the BGP re-convergency, RIB and FIB update. BGP convergency time is longer than IGP. The document [BGP-Converge-Time1] has shown that the BGP convergence time varies from 50sec to couple of hundred seconds. The analysis [BGP-Converge-Time2] indicated that per entry update takes about 150us, and it takes o(75s) for 500k routes, or o(150s) for 1M routes.

P4. More frequent IGP flooding and BGP update in whole satellite network

To provide the global coverage, a satellite constellation will have many ground stations deployed. For example, StarLink has applied for the license for up to one million ground stations [StarLink-Ground-Station-Fcc], in which, more than 50 gateway ground stations (equivalent to the PE_GS2) have been registered [SpaceX-Ground-Station-Fcc] and deployed in U.S. [StarLink-GW-GS-map]. It is expected that the gateway ground station will grow quickly to couple of thousands [Tech-Comparison-LEOs]. This means almost each satellite in the satellite network would have a ground station connected. , Due to the fact that all satellites are moving, many IGP adjacency changes may occur in a shorter period of time described in Appendix B.1 and result in the problem P1 and P2 constantly occur.

P5. Service is not steady

Due to the problems P1 to P3, the service provider of satellite constellation is hard to provide a steady service for broadband service by using inter-satellite network and traditional routing technologies.

As a summary, the traditional routing technology is problematic for large scale inter-satellite networking for Internet. Enhancements on traditional technologies, or new technologies are expected to solve the specific issues associated with satellite networking.

6. IANA Considerations

This memo includes no request to IANA.

7. Security Considerations

Security considerations for communication between satellite and ground station, or between satellites are described in corresponding sections. There is no extra security issue introduced by this memo.

8. Contributors

9. Acknowledgements

10. References

10.1. Normative References

[RFC4271] Rekhter, Y., Ed., Li, T., Ed., and S. Hares, Ed., "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, DOI 10.17487/RFC4271, January 2006, <https://www.rfc-editor.org/info/rfc4271>.

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

[RFC7142] Shand, M. and L. Ginsberg, "Reclassification of RFC 1142 to Historic", RFC 7142, DOI 10.17487/RFC7142, February 2014, <https://www.rfc-editor.org/info/rfc7142>.

[RFC2453] Malkin, G., "RIP Version 2", STD 56, RFC 2453, DOI 10.17487/RFC2453, November 1998, <https://www.rfc-editor.org/info/rfc2453>.

[RFC7868] Savage, D., Ng, J., Moore, S., Slice, D., Paluch, P., and R. White, "Cisco’s Enhanced Interior Gateway Routing Protocol (EIGRP)", RFC 7868, DOI 10.17487/RFC7868, May 2016, <https://www.rfc-editor.org/info/rfc7868>.

10.2. Informative References

[KeplerianElement] "Keplerian elements", <https://en.wikipedia.org/wiki/Orbital_elements>.

[GEO-Coverage] "Coverage of a geostationary satellite at Earth", <https://www.planetary.org/space-images/coverage-of-a-geostationary>.

[Nttdocomo-6G] "NTTDPCOM 6G White Paper", <https://www.nttdocomo.co.jp/english/binary/pdf/corporate/technology/whitepaper_6g/DOCOMO_6G_White_PaperEN_20200124.pdf>.

[ITU-6G] "ITU 6G vision", <https://www.itu.int/dms_pub/itu-s/opb/itujnl/S-ITUJNL-JFETF.V1I1-2020-P09-PDF-E.pdf>.

[Surrey-6G] "Surrey 6G vision", <https://www.surrey.ac.uk/sites/default/files/2020-11/6g-wireless-a-new-strategic-vision-paper.pdf>.

[OSI-Model]
"OSI Model", <https://en.wikipedia.org/wiki/OSI_model>.

[StarLink] "Star Link", <https://en.wikipedia.org/wiki/Starlink>.

[China-constellation]
"China Constellation", <https://www.itu.int/ITU-R/space/asreceived/Publication/DisplayPublication/23706>.

[ESA-HydRON]
"HydRON: Fiber in the sky", <https://www.esa.int/ESA_Multimedia/Videos/2021/04/HydRON_Fibre_in_the_sky>.

[SpaceX-Non-GEO]
"FCC report: SPACEX V-BAND NON-GEOSTATIONARY SATELLITE SYSTEM", <https://fcc.report/IBFS/SAT-LOA-20170301-00027/1190019.pdf>.

[Satellite-coverage]
Alan R.Washburn, Department of Operations Research, Naval Postgraduate School, "Earth Coverage by Satellites in Circular Orbit", <https://faculty.nps.edu/awashburn/Files/Notes/EARTHCOV.pdf>.

[Microwave-vs-Laser-communication]
International Journal for Research in Applied Science and Engineering Technology (IJRASET), "Comparison of Microwave and Optical Wireless Inter-Satellite Links", <https://www.ijraset.com/fileserve.php?FID=7815>.

[Laser-communication-range]
"Interferometric optical communications can potentially lead to robust, secure, and naturally encrypted long-distance laser communications in space by taking advantage of the underlying physics of quantum entanglement.", <https://www.laserfocusworld.com/optics/article/16551652/interferometry-quantum-entanglement-physics-secures-spacetospace-interferometric-communications>.

[BGP-Table-Size]
"BGP in 2020 - BGP table", <https://blog.apnic.net/2021/01/05/bgp-in-2020-the-bgp-table/>.

[BGP-Converge-Time1]
"BGP in 2020 - BGP Update Churn",
<https://labs.apnic.net/?p=1397>.

[BGP-Converge-Time2]
"Bringing SDN to the Internet, one exchange point at the time",
<https://www.cs.princeton.edu/courses/archive/fall14/cos561/docs/SDX.pdf>.

[StarLink-Ground-Station-Fcc]
"APPLICATION FOR BLANKET LICENSED EARTH STATIONS",
<https://fcc.report/IBFS/SES-LIC-INTR2019-00217/1616678>.

[SpaceX-Ground-Station-Fcc]
"List of SpaceX applications for ground stations",
<https://fcc.report/IBFS/Company/Space-Exploration-Technologies-Corp-SpaceX>.

[Tech-Comparison-LEOs]
"A Technical Comparison of Three Low Earth Orbit Satellite Constellation Systems to Provide Global Broadband",
<http://www.mit.edu/~portillo/files/Comparison-LEO-IAC-2018-slides.pdf>.

[StarLink-GW-GS-map]
"StarLink gateway ground station map",
<https://www.google.com/maps/d/u/0/viewer?mid=1H1x8jZs8vfjy60TvKgpbYs_grargieVw>.

[UCL-Mark-Handley]
"Using ground relays for low-latency wide-area routing in megaconstellations",
<https://discovery.ucl.ac.uk/id/eprint/10090242/1/hotnets-ucl.pdf>.

[TR-23.700-27]
3GPP, "Study on support of satellite backhauling in 5GS", December 2022.

[TR-23.700-28]
3GPP, "Study on 5GC enhancement for satellite access Phase 2", December 2022.

[TS-23.501]
3GPP, "System architecture for the 5G System (5GS)", December 2021.

[TS-23.502]
3GPP, "Procedures for the 5G System (5GS)", December 2021.

[CCSDS] CCSDS, "Consultative Committee for Space Data Systems", <https://public.ccsds.org/default.aspx>.

[DVB] DVB, "Digital Video Broadcasting", <https://en.wikipedia.org/wiki/DVB>.

Appendix A. Basics of Satellite Constellation

This section will introduce some basics for satellite such as orbit parameters, coverage estimation, minimum number of satellite and orbit plane required, real deployments.

A.1. Satellite Orbit

The orbit of a satellite can be either circular or ecliptic, it can be described by following Keplerian elements [KeplerianElement]:

1. Inclination (i)

2. Longitude of the ascending node (Omega)

3. Eccentricity (e)

4. Semimajor axis (a)

5. Argument of periapsis (omega)

6. True anomaly (nu)

For a circular orbit, two parameters, Inclination and Longitude of the ascending node, will be enough to describe the orbit.

A.2. Coverage of LEO and VLEO Satellites and Minimum Number Required

The coverage of a satellite is determined by many physical factors, such as spectrum, transmitter power, the antenna size, the altitude of satellite, the air condition, the sensitivity of receiver, etc. EIRP could be used to measure the real power distribution for coverage. It is not deterministic due to too many variants in a real environment. The alternative method is to use the minimum elevation angle from user terminals or gateways to a satellite. This is easier and more deterministic. [SpaceX-Non-GEO] has suggested originally the minimum elevation angle of 35 degrees and deduced the radius of the coverage area is about 435km and 1230km for VLEO (altitude 335.9km) and LEO (altitude 1150km) respectively. The details about

how the coverage is calculated from the satellite elevation angle can be found in [Satellite-coverage].

Using this method to estimate the coverage, we can also estimate the minimum number of satellites required to cover the earth surface.

It must be noted, SpaceX has recently reduced the required minimum elevation angle from 35 degrees to 25 degrees. The following analysis still use 35 degrees.

Assume there is multiple orbit planes with the equal angular interval across the earth surface (The Longitude of the ascending node for sequential orbit plane is increasing with a same angular interval). Each orbit plane will have:

1. The same altitude.

2. The same inclination of 90 degree.

3. The same number of satellites.

With such deployment, all orbit planes will meet at north and south pole. The density of satellite is not equal. Satellite is more dense in the space above the polar area than in the space above the equator area. Below estimations are made in the worst covered area, or the area of equator where the satellite density is the minimum.

Figure 12 illustrates the coverage area on equator area, and each satellite will cover one hexagon area. The figure is based on plane geometry instead of spherical geometry for simplification, so, the orbit is parallel approximately.

Figure 13 shows how to calculate the radius (Rc) of coverage area from the satellite altitude (As) and the elevation angle (b).

```
                        x
                 |      |
                 |      |
                 x   *********
                 |   *    |    *
                 |  *     |     *
            *********   x      *
           *     |  *    |     *
          *      |   *   |    *
        *        x   *********
         *       |   *   |
          *      |  *    |
           *********   x
                 |      |
                 |   orbit 2          ^ north
                 x                    |
                 |                    |
                 |                    |
             orbit 1                  +-------> east
```

Figure 12: Satellite coverage on ground

```
                |<---  2*Rc --->|

                        + Satellite
                       /|
                      / |
                     /  |
                    / b |
                   /-\  +
                  /   * |     __Earth surface
                 /  *   |    /
                / *_----+----__
               +                +
                *              *
                 *            *
                  *   2*a   *
                   *  ___  *
                    *_   _*
                     *   *
                      * *
                       * Earth center
```

Figure 13: Satellite coverage estimation

x The vertical projection of satellite to Earth

Re The radius of the Earth, Re=6378(km)

As The altitude of a satellite

Rc The radius (arc length) of the coverage, or, the arc length of hexagon center to its 6 vertices. Rc=Re*(a*pi)/180

a The cap angle for the coverage area (the RC arc). a = arccos((Re/(Re+As))*cos(b))-b.

b The least elevation angle that a ground station or a terminal can communicate with a satellite, b = 35 degree.

Ns The minimum number of satellites on one orbit plane, it is equal to the number of the satellite's vertical projection on Earth, so, Ns = 180/(a*cos(30))

No The minimum number of orbit (with same inclination), it is equal to the number of the satellite orbit's vertical projection, so, No = 360/(a*(1+sin(30)))

For a example of two type of satellite LEO and VEO, the coverages are calculated as in Table 1:

Parameters	VLEO1	VLEO2	LEO1	LEO2
As(km)	335.9	450	1100	1150
a(degree)	3.907	5.078	10.681	11.051
Rc(km)	435	565	1189	1230
Ns	54	41	20	19
No	62	48	23	22

Table 1: Satellite coverage estimation for LEO and VLEO examples

A.3. Real Deployment of LEO and VLEO for Satellite Network

Obviously, the above orbit parameter setup is not optimal since the sky in the polar areas will have the highest density of satellite.

In the real deployment, to provide better coverage for the areas with denser population, to get redundance and better signal quality, and to make the satellite distance within the range of inter-satellite communication (2000km [Laser-communication-range]), more than the minimum number of satellites are launched. For example, different orbit planes with different inclination/altitude are used.

Normally, all satellites are grouped by orbit planes, each group has a number of orbit planes and each orbit plane has the same orbit parameters, so, each orbit in the same group will have:

1. The same altitude

2. The same inclination, but the inclination is less than 90 degrees. This will result in the empty coverage for polar areas and better coverage in other areas. See the orbit picture for phrase 1 for [StarLink].

3. The same number of satellites

4. The same moving direction for all satellites

The proposed deployment of SpaceX can be seen in [SpaceX-Non-GEO] for StarLink.

The China constellation deployment and orbit parameters can be seen in [China-constellation].

Appendix B. Communications for Satellite Constellation

Unlike the communication on ground, the communication for satellite constellation is much more complicated. There are two mobility aspects, one is between ground-station and satellite, another is between satellites.

In the traditional mobility communication system, only terminal is moving, the mobile core network including base station, front haul and back haul are static, thus an anchor point, i.e., PGW in 4G or UPF in 5G, can be selected for the control of mobility session. Unfortunately, when satellite constellation joins the static network system of Internet on ground, there is no such anchor point can be selected since the whole satellite constellation network is moving.

Another special aspect that can impact the communication is that the fast moving speed of satellite will cause frequent changes of communication peers and link states, this will make big challenges to the network side for the packet routing and delivery, session control and management, etc.

B.1. Dynamic Ground-station-Satellite Communication

All satellites are moving and will lead to the communication between ground station and satellite can only last a certain period of time. This will greatly impact the technologies for the satellite networking. Below illustrates the approximate speed and the time for a satellite to pass through its covered area.

In Table 2, VLEO1 and LEO3 have the lowest and highest altitude respectively, VLEO2 is for the highest altitude for VLEO. We can see that longest communication time of ground-station-satellite is less than 400 seconds, the longest communication time for VLEO ground-station-satellite is less than 140 seconds.

The "longest communication time" is for the scenario that the satellite will fly over the receiver ground station exactly above the head, or the ground station will be on the diameter line of satellite coverage circular area, see Figure 12.

Re The radius of the Earth, Re=6378(km)

As The altitude of a satellite

AL The arc length(in km) of two neighbor satellite on the same orbit plane, AL=2*cos(30)*(Re+As)*(a*pi)/180

SD The space distance(in km) of two neighbor satellite on the same orbir plane, SD=2*(Re+As)*sin(AL/(2*(Re+As))).

V the velocity (in m/s) of satellite, V=sqrt(G*M/(Re+As))

G Gravitational constant, G=6.674*10^(-11)(m^3/(kg*s^2))

M Mass of Earth, M=5.965*10^24 (kg)

T The time (in second) for a satellite to pass through its cover area, or, the time for the station-satellite communication. T= ALs/V

Parameters	VLEO1	VLEO2	LEO1	LEO2	LEO3
As(km)	335.9	450	1100	1150	1325
a(degree)	3.907	5.078	10.681	11.051	12.293
AL(km)	793	1048	2415	2515	2863
SD(km)	792.5	1047.2	2404	2503.2	2846.1
V(km/s)	7.7	7.636	7.296	7.272	7.189
T(s)	103	137	331	346	398

Table 2: The time for the ground-station-satellite communication

B.2. Dynamic Inter-satellite Communication

B.2.1. Inter-satellite Communication Overview

In order to form a network by satellites, there must be an inter-satellite communication. Traditionally, inter-satellite communication uses the microwave technology, but it has following disadvantages:

1. Bandwidth is limited and only up to 600M bps [Microwave-vs-Laser-communication].

2. Security is a concern since the microwave beam is relatively wide and it is easy for 3rd party to sniff or attack.

3. Big antenna size.

4. Power consumption is high.

5. High cost per bps.

Recently, laser is used for the inter-satellite communication, it has following advantages, and will be the future for inter-satellite communication.

1. Higher bandwidth and can be up to 10G bps [Microwave-vs-Laser-communication].

2. Better security since the laser beam size is much narrower than microwave, it is harder for sniffing.

3. The size of optical lens for laser is much smaller than microwave's antenna size.

4. Power saving compared with microwave.

5. Lower cost per bps.

The range for satellite-to-satellite communications has been estimated to be approximately 2,000 km currently [Laser-communication-range].

From Table 2, we can see the Space Distance (SD) for some LEO (altitude over 1100km) are exceeding the celling of the range of laser communication, so, the satellite and orbit density for LEO need to be higher than the estimation values in the Table 1.

Assume the laser communication is used for inter-satellite communication, then we can analyze the lifetime of inter-satellite communication when satellites are moving. The Figure 14 illustrates the movement and relative position of satellites on three orbits. The inclination of orbit planes is 90 degrees.

```
                    + North pole
                   /|\
                  | s |
                 s  |  s
                /   s   \
                s   |   s
                |   s1  |
                s4  |   s6
                |   s2  |   -------- Equator
                s5  |   s7
                |   s3  |
                s   |   s
                \   s   /
                 s  |  s
                  | s |
                   \|/
                    +  South pole
```

Figure 14: Satellite movement

There are four scenarios:

1. For satellites within the same orbit
   The satellites in the same orbit will move to the same direction with the same speed, thus the interval between satellites is relatively steady. Each satellite can communicate with its front and back neighbor satellite as long as satellite's orbit is maintained in its life cycle. For example, in Figure 14, s2 can communication with s1 and s3.

2. For satellites between neighbor orbits in the same group at non-polar areas
   The orbits for the same group will share the same orbit altitude and inclination. So, the satellite speed in different orbit are also same, but the moving direction may be same or different. Figure 15 illustrates this scenario. When the moving direction is the same, it is similar to the scenario 1, the relative position of satellites in different orbit are relatively steady as long as satellite's orbit is maintained in its life cycle. When the moving direction is different, the relative position of satellites in different orbit are un-steady, this scenario will be analyzed in more details in Appendix B.2.2.

3. For satellites between neighbor orbits in the same group at polar areas
   For satellites between neighbor orbits with the same speed and moving direction, the relative position is steady as described in #2 above, but the steady position is only valid at areas other than polar area. When satellites meet in the polar area, the relative position will change dramatically. Figure 16 shows two satellites meet in polar area and their ISL facing will be swapped. So, if the range of laser pointing angle is 360 degrees and tracking technology supports, the ISL will not be flipping after passing polar area; Otherwise, the link will be flipping and inter-satellite communication will be interrupted.

4. For satellites between different orbits in the different group
   The orbits for the different group will have different orbit altitude, inclination and speed. So, the relative position of satellite is not static. The inter-satellite communication can only last for a while when the distance between two satellite is within the limit of inter-satellite communication, that is 2000km for laser [Laser-communication-range], this scenario will be analyzed in more details in Appendix B.2.3

```
                 i+N/2         i+1+N/2       i+2+N/2
               / \           / \           / \
              /   \         /   \         /   \
             S1    \       S2    \       S3    \
            /       S4    /      S5     /      S6
           /         \   /         \   /         \
          /           \ /           \ /           \
          i-1           i             i+1
```

* The total number of orbit planes are N
* The number (i-1, i, i+1,...) represents the Orbit index
* The bottom numbers (i-1, i, i+1) are for orbit planes on which satellites (S1, S2, S3) are moving from bottom to up.
* The top numbers (i+N/2, i+1+N/2, i+2+N/2) are for orbit planes on which satellites (S4, S5, S6) are moving from up to bottom.

Figure 15: Two satellites with same altitude and inclination (i) move in the same or opposite direction

```
\      /
 P3   P4
  \  /
   \/
   /\
  /  \
 P1   P2
/      \
```

* Two satellites S1 and S2 are at position P1 and P2 at time T1
* S1's right facing ISL connected to S2's left facing ISL
* S1 and S2 move to the position P4 and P3 at time T2
* S1's left facing ISL connected to S2's right facing ISL

Figure 16: Two satellites meeting in the polar area will change its facing of ISL

B.2.2. Satellites on Adjacent Orbit Planes with Same Altitude

For satellites on different orbit planes with same altitude, the estimation of the lifetime when two satellite can communicate are as follows.

Figure 17 illustrates a general case that two satellites move and intersect with an angle A.

```
                                        ^ V2
                                       /
                                      /
                                    +-
                                   /  \ A
                     -------------+----+----> V1
                                 /
                                /
```

Figure 17: Two satellites (speed vector V1 and V2) intersect with angle A

More specifically, for orbit planes with the inclination angle i, Figure 18 illustrates two satellites move in the opposite direction and intersect with an angle 2*i.

```
                                 ^ move from south to north
                            \      /
                             \    /
                              \  /\
                               \/  | A = 2*i
                               /\  |
                              /  \/
                             /    \
                            /      V move from north to south
```

Figure 18: Two satellites with same altitude and inclination (i) intersect with angle A=2*i

Follows are the math to calculate the lifetime of communication. Table 3 are the results using the math for two satellites with different altitudes and different inclination angles.

Dl The laser communication limit, Dl=2000km [Laser-communication-range]

A The angle between two orbit’s vertical projection on Earth. A=2*i

V1 The speed vector of satellite on orbit1

V2 The speed vector of satellite on orbit2

|V| the magnitude of the difference of two speed vector V1 and V2, |V|=|V1-V2|=sqrt((V1-V2*cos(A))^2+(V2*sin(A))^2). For satellites with the same altitude and inclination angle i, V1=V2, so, |V|=V1*sqrt(2-2*cos(2*i))=2V1*sin(i)

T The lifetime two satellites can communicate, or the time of two satellites' distance is within the range of communication, T = 2*Dl/|V|.

i (degree)	80	80	65	65	50	50
Alt (km)	500	800	500	800	500	800
|V| (km/s)	14.98	14.67	13.79	13.5	11.66	11.41
T(s)	267	273	290	296	343	350

Table 3: The lifetime of communication for two LEOs (with two altitudes and three inclination angles)

B.2.3. Satellites on Adjacent Orbit Planes with Different Altitude

For satellites on different orbit planes with different altitude, the estimation of the lifetime when two satellite can communicate are as follows.

Figure 19 illustrates two satellites (with the altitude difference Da) move and intersect with an angle A.

```
                                   ^ V2
                                  /
                                 /
                        -------+  /
                        Da    /| +-
                             / |/  \ A
                    ----------/--+----+----> V1
                              /  /
                                /
                               /
                              /
```

Figure 19: Satellite (speed vector V1 and V2, Altitude difference Da) intersects with Angle A

Follows are the math to calculate the lifetime of communication

Dl The laser communication limit, Dl=2000km [Laser-communication-range]

Da Altitude difference (in km) for two orbit planes

A The angle between two orbit's vertical projection on Earth

Vl The speed vector of satellite on orbit 1

V2 The speed vector of satellite on orbit 2

|V| the magnitude of the difference of two speed vector V1 and V2, |v|=|V1-V2|=sqrt((V1-V2*cos(A))^2+(V2*sin(A))^2)

T The lifetime two satellites can communicate, or the time of two satellites' distance is within the range of communication, T = 2*sqrt(Dl^2-Da^2)/|V|

Using formulas above, below is the estimation for the life of communication of two satellites when they intersect. Table 4 and Table 5 are for two VLEOs with the difference of 114.1km for altitude. (VLEO1 and VLEO2 on Table 2). Table 6 and Table 7 are for two LEOs with the difference of 175km for altitude (LEO2 and LEO3 on Table 2).

Parameters	VLEO1	VLEO2
As(km)	335.9	450
V (km/s)	7.7	7.636

Table 4: Two VLEO with different altitude and speed

A (degree)	0	10	45	90	135	180
|V| (km/s)	0.065	1.338	5.869	10.844	14.169	15.336
T(s)	61810	2984	680	368	282	260

Table 5: Two VLEO intersects with different angle and the life of communication

Parameters	LEO1	LEO2
As(km)	1150	1325
V (km/s)	7.272	7.189

Table 6: Two LEO with different altitude and speed

A (degree)	0	10	45	90	135	180
|V| (km/s)	0.083	1.263	5.535	10.226	13.360	14.461
T(s)	47961	3155	720	390	298	276

Table 7: Two LEO intersects with different angle and the life of communication

Appendix C. Change Log

* Initial version, 07/03/2021

* 01 version, 10/20/2021

* 02 version, 2/13/2022

* 03 version, 7/5/2022

* 04 version, 1/4/2023

* 05 version, 7/5/2023

* 06 version, 1/3/2024

Authors' Addresses

Lin Han (editor)
Futurewei Technologies, Inc.
2330 Central Expy
Santa Clara, CA 95050,
United States of America
Email: lhan@futurewei.com

Richard Li
Futurewei Technologies, Inc.
2330 Central Expy
Santa Clara, CA 95050,
United States of America
Email: rli@futurewei.com

Alvaro Retana
Futurewei Technologies, Inc.
2330 Central Expy
Santa Clara, CA 95050,
United States of America
Email: alvaro.retana@futurewei.com

Meiling Chen
China Mobile
32, Xuanwumen West
BeiJing 100053
China
Email: chenmeiling@chinamobile.com

Li Su
China Mobile
32, Xuanwumen West
BeiJing 100053
China
Email: suli@chinamobile.com

Tianji Jiang
China Mobile
1525 McCathy Blvd.
Milpitas, CA 95035,
United States of America
Email: tianjijiang@chinamobile.com

Network Working Group
Internet-Draft
Intended status: Informational
Expires: 4 March 2024

L. Han, Ed.
A. Retana
R. Li
Futurewei Technologies, Inc.

1 September 2023

Semantic Address Based Instructive Routing for Satellite Network
draft-lhan-satellite-instructive-routing-03

Abstract

This document presents a method to do IP routing over satellite network that consists of LEO (Low Earth Orbit) satellites and ground-stations. The method uses the source routing mechanism. The whole routing info is obtained by path calculation. The routing path information is converted to be a list of instructions and embedded into user packet's IPv6 extension header. At each hop or each satellite, the routing process engine will forward the packet based on the specified instruction for the satellite. Until the packet reaches the edge of satellite network, or the last satellite, the packet will be sent to a ground station.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

Massive LEO constellation is expected to be used for future Internet. It has raised challenges to the current IP networking technologies to support such super-fast-moving network. [I-D.lhan-problems-requirements-satellite-net] has analyzed the problems when using the regular routing protocols in such network.

Since all satellites in a LEO constellation are well organized and form a kind of multi-layered grid network, each satellite's relative position in the satellite network will be steady during its life time. [I-D.lhan-satellite-semantic-addressing] has proposed to use couple of indexes to identify each satellite in the network. The combination of the indexes is called the satellite semantic address. The semantic address can be embedded into the field of the interface identifier (i.e., the rightmost 64 bits) of the IPv6 address, if IPv6 is used in the satellite network.

This memo proposes a method for routing for LEO satellite network, it is based on the satellite semantic address. It is a source routing mechanism and conceptually similar to SRv6 (IPv6 Segment Routing) [RFC8754] with loose-hop, but with many differences in the architecture and details. The routing information is embedded into the IPv6 packet as routing extension header defined in [RFC8200]. Unlike the SRv6 [RFC8754] and programming [RFC8986], The new method will not use IPv6 SID (Segment Identifier) to represent the segments on the routing path. Instead, it will convert the segments on the path to be a list of instructions since each satellite could be represented by the semantic address. Each instruction can tell each satellite how to forward the packet to an adjacent satellite and when to stop, either on the same orbit, or on the adjacent orbit.

Compared with the traditional IP forwarding, the new method will not use TCAM (Ternary Content-addressable Memory) lookup for IP prefix. Each satellite only needs to store a simple adjacency table. Therefore, the new method can save significant TCAM and the processing time for routing/forwarding tables.

It must be noted this memo just describes one aspect of the whole solution for satellite constellation used for Internet access and NTN (Non-Terrestrial Network) integration with 5G, following areas are not covered in this memo and will be addressed in other documents separately:

1. IP forwarding path determination for a LEO constellation. There are different strategies and algorithms to determine the IP path. One example using modified OSPF and Dijkstra algorithm [I-D.retana-lsr-ospf-monitor-node] to get the shortest geographic path can be found in [Large-Scale-LEO-Network-Routing].

2. Data planes for different scenarios, such as Internet access and NTN integration.

3. Other protocols for control plane.

2. Terminology

LEO	Low Earth Orbit with the altitude from 180 km to 2000 km.
LEO constellation	LEO constellation consists of certain number of LEOs. Each LEO has pre-assigned orbit element.
ISL	Inter Satellite Link
GS	Ground Station, a device on ground connecting satellite. In the document, GS will hypothetically provide L2 and/or L3 functionality in addition to process/transmit/receive radio wave. It might be different as the reality that the device to process/transmit/receive radio wave and the device to provide L2 and/or L3 functionality could be separated.
L2	Layer 2, or Data Link Layer in OSI model [OSI-Model]
L3	Layer 3, or Network Layer in OSI model [OSI-Model], it is also called IP layer in TCP/IP model
OS	Operating System
NTN	Non-Terrestrial Network
SID	Segment Identifier
Sat-GS Links	Wireless links between satellites and ground-stations, it consists of uplink (from ground to satellite) and downlink (from satellite to ground.
Link Metrics	The cost of the outgoing interface for routing,

typically, it may indicate the bandwidth, delay or other costs for the interface.

Sat_ID Satellite Index, the Index for the satellite in a orbit plane, see [I-D.lhan-satellite-semantic-addressing]

Obp_ID Orbit Plane Index, the Index for the orbit plane in a shell group of satellite, see [I-D.lhan-satellite-semantic-addressing]

Shl_ID Shell Index, the Index for the shell group of satellite in a satellite constellation, see [I-D.lhan-satellite-semantic-addressing]

Intf_ID Interface Index

Sat_Addr Satellite Semantic Address, it consists of indexes Shl_ID, Obp_ID and Sat_ID. It is 32-bit long and is defined in Section 5.4 in [I-D.lhan-satellite-semantic-addressing]

Sat_MacAddr The MAC (Media Access Control) Address for a satellite

3. Review of LEO satellite constellation for future Internet

LEO satellite constellation is expected to be integrated with terrestrial network in future Internet. StarLink project [StarLink] has launched its satellites and provided the beta service in some areas. 3GPP [ThreeGPP] has studied the issues when NTN is integrated with Internet and 5G. 3GPP [TR38-821] has also proposed the Satellite-based NG-RAN architectures for NTN integration. In the 3GPP new Release 18 (in-progress), there is a working item "Study on 5G System with Satellite Backhaul" [TR23-700]. In which, LEO satellite network will provide the transport functionality for 5G RAN access network. As a summary, the targets of LEO constellation for future Internet and NTN integration are as follows:

1. Global coverage: The Satellite network should cover all places on earth and any flying objects as long as the place or objects are below LEO attitude and within the coverage footprint of satellite constellation, the satellite network should be the complementary to terrestrial network.

2. Internet access: The Satellite network can provide the Internet access service for covered areas.

3. NTN integration: The Satellite network is fully integrated with Internet including Wireless such as 4G or 5G.

4. Competitive service: The Satellite network can provide the services that are competitive to terrestrial network in terms of service stability, Quality of Service, especially the latency for Satellite network is shorter.

As a new form of network, LEO constellation has lots of difference with the steady terrestrial network especially in the mobility. [I-D.lhan-problems-requirements-satellite-net] has analyzed the movement and coverage of satellite. For a massive LEO constellation, all satellites are moving on the allocated orbits, and form one or multiple layers of network. Finally, the massive LEO constellation will have the following unprecedented mobility:

1. Each LEO moves at the speed of 7.x km/s.

2. Ground Stations move at the speed of 463 m/s due to earth rotation.

3. Half of LEOs move on the direction that is different with another half of LEOs.

4. Huge number of links between satellites and ground-stations, and all of them are constantly flipping within short period of time. All Link Metrics of Sat-GS Links are also constantly changing.

5. All Link Metrics of ISL on the Longitude direction are constantly changing.

6. All Links of ISL on the Longitude direction may be interrupted at two polar areas.

7. All Link Metrics of ISL on the radius direction (for satellites with different altitude) are constantly changing.

8. All Links of ISL on the radius direction can only last for a limited time.

4. Basics of Instructive Routing

In IP routing or forwarding, the IP path consists of a list of IP nodes (hops). In LEO satellite network, the IP forwarding path is a list of satellites. Instructive routing essentially is a mechanism that converts satellites on the path to a list of segment and then to a list of instructions. It will utilize the special characters of LEO satellite network to achieve the minimized packet overhead while

the forwarding functions can be executed quickly.

A typical LEO satellite network is an interleaved and meshed network moving constantly. Each satellite only has limited adjacent satellites, thus the limited packet forwarding directions (see Section 4.1).

The satellites on a forwarding path can be converted to a list of segments. The number of segments is normally much smaller than the number of satellites on the path.

The number of segment type will determine the number of instruction type. Since the segment type is also limited (see Section 4.2), so the instruction type is limited.

Finally, combining the above characters and with the use of semantic address, the Instructive Routing will only introduce limited overhead that is much smaller than SRv6 and SRv6 with compressed SID.

4.1. Forwarding Directions

When using ISL for satellites in a LEO constellation, each layer of network will have satellite nodes connected by limited ISLs. A typical satellite will have about six ISL to connected to its adjacent satellites in 3D space. Additionally, there might have very few numbers of ISL working as un-steady link to connect to other satellites. Un-stead links are those between satellites moving to different directions, see [I-D.lhan-problems-requirements-satellite-net] for the detailed explanation. After using the semantic address for each satellite, the satellite relationship will be static. Figure 1 illustrates one satellite and its six direct connected adjacent satellites, it is easy to determine some indexes of its adjacent satellites:

1. S0, S1 and S2 have the same Shl_ID, the difference of Obp_ID between S0 and S1, S0 and S2 are both equal to one.

2. S0, S3 and S4 have the same Shl_ID and Obp_ID, the difference of Sat_ID between S0 and S3, S0 and S4 are both equal to one.

3. S0, S5 and S6 have different Shl_ID, and the difference of Shl_ID between S0 and S5, S0 and S6 are both equal to one.

Another benefit to use the semantic address is that the packet forwarding for routing and switching will be simplified significantly. There will be only six major forwarding directions to the directly connected adjacent satellites described above, plus one or few specified directions probably. The specified direction is to

forward packet to a specified adjacent satellite through an un-steady link. The un-steady link can connect to any satellite but only last for a short time. The usages of un-steady links are expected to be limited and are not major scenarios in a LEO constellation. Following are all directions for forwarding:

1. Forward to the Sat_ID Incremental or Decremental directions.

2. Forward to the Obp_ID Incremental or Decremental directions.

3. Forward to the Shl_ID Incremental or Decremental directions.

4. Forward to a specified satellite through an un-steady link.

```
                    ^ Shl_ID Incremental direction
                    |
                      /
                     /
                    S5    ^ Sat_ID Increment direction
                   /|    /
                  / |   S3
               / /  |  /       /
              / /   | /       /
             /      |/       /
            S2------S0------S1  -> Obp_ID Increment direction
           /       /|   /  /
          /       / |  /  /
         /       /  | /  /
                S4  |/
                    S6
                   /
                  /
                 /
```

Figure 1: The LEO Satellite Relationship in 3D Space

4.2. Forwarding Segments

A forwarding segment is defined as a list of satellites, and four type segments are defined for LEO satellite network where semantic address is used:

1. Segment with adjacent Shl_ID: For any direct adjacent satellites on the segment, their Shl_ID are also adjacent (differ by one).

2. Segment with adjacent Obp_ID: For any direct adjacent satellites on the segment, their Obp_ID are also adjacent (differ by one), the Shl_ID are the same.

3. Segment with adjacent Sat_ID: For any direct adjacent satellites on the segment, their Sat_ID are also adjacent (differ by one), the Obp_ID and Shl_ID are identical.

4. Segment with non-adjacent index: this segment only has two satellites and two satellites do not belong to the above three categories.

4.3. Forwarding Instructions

Each forwarding instruction consists of Functional Code and Argument (see Section 6). For the most often used instructions, the Argument represents one specified index (Sat_ID or Obp_ID or Shl_ID) of a satellite semantic address and only has the size of one octet.

Each segment maps to a forwarding instruction that can guides the packet forwarded at each satellite from the start to the end of the segment. For the segment types (1) to (3) described in Section 4.2, there are two directions to forward packet, each direction can be defined as either an increment or a decrement of a specified index. For type (4), there is one direction to forward packet. In total we have seven directions to forward packets among all satellites: to the satellite ahead or behind; to either sides; above or below; or to another non-adjacent satellite.

When an IP packet is forwarded on a segment by an instruction, at each satellite, the forwarding logic needs to check if the packet reaches the end of the segment. In the regular segment routing, the long size of SID is used to do such indication. But for satellite network, since 32-bit satellite’s semantic address is embedded into the IPv6 address, it is not needed to include the long SID into the packet header. Instead, we only need to compare one octet index of the current satellite’s semantic address, instead of whole IPv6 address, with the Argument in the instruction.

4.4. Example

Figure 2 illustrates a 2D example. It shows how a packet is forwarded in a grid satellite network. Intuitively, we can obtain the list of instructions to guide the packet and get the forwarding behaviors at different satellites. Following is an example:

1. At S1 to S2, forward packet to the Sat_ID Incremental direction, until the packet reaches S2

2. At S2 to S3, forward packet to the Obp_ID Incremental direction, until the packet reaches the orbit plane of S3

3. At S3 to S4, forward packet to the Sat_ID Incremental direction, until the packet reaches S4

4. At S4 to S5, forward packet to the Obp_ID Decremental direction, until the packet reaches the orbit plane of S5

5. At S5 to S6, forward packet to the Sat_ID Decremental direction, until the packet reaches S6

By using a specified index of semantic address as the argument as described in Section 4.3, we can further simplify the above instructions as:

1. At S1 to S2, forward packet to the Sat_ID Incremental direction, until the packet reaches a satellite and the satellite's Sad_ID is equal to the given instruction argument (S2's Satellite Index)

2. At S2 to S3, forward packet to the Obp_ID Incremental direction, until the packet reaches a satellite and the satellite's Obp_ID is equal to the given instruction argument (S3's Orbit Plane Index)

3. At S3 to S4, forward packet to the Sat_ID Incremental direction, until the packet reaches a satellite and the satellite's Sat_ID is equal to the given instruction argument (S4's Satellite Index)

4. At S4 to S5, forward packet to the Obp_ID Decremental direction, until the packet reaches a satellite and the satellite's Obp_ID is equal to the given instruction argument (S5's Orbit Plane Index)

5. At S5 to S6, forward packet to the Sat_ID Decremental direction, until the packet reaches a satellite and the satellite's Sat_ID is equal to the given instruction argument (S6's Satellite Index)

```
               ^ Sat_ID Incremental Direction
               |
               |
               +----> Obp_ID Incremental Direction

               x: The ISL is down

                    Obp_ID Obp_ID+1 Obp_ID+2 Obp_ID+3 Obp_ID+4
                       |       |       |       |       |
                   ----+-------S5<<<<<<S<<<<<<<S4------+----
                       |       V       x       ^       |
                       |       V       |       ^       |
                   ----+-------S6------+---x---S-------+----
                       |       |       |       ^       |
                       x       x       x       ^       |
                   ----S2>>>>>>S>>>>>>>S>>>>>>>S3------+----
                       ^       |       |       |       |
                       ^       |       |       |       |
                   ----S---x---+-------+-------+-------+----
                       ^       |       |       |       |
                       ^       |       |       |       |
                   ----S1--x---+-------+-------+-------+----
                       |       |       |       |       |
                       |       |       |       |       |
```

Figure 2: Packet Forwarding in 2D LEO satellite constellation network

5. IPv6 Routing Header for Instructive Routing

For instructive routing, IPv6 routing header is used with a new routing type "Instructive Routing Type". The format of the new routing header is illustrated in Figure 3.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Next Header   |  Hdr Ext Len  | Routing Type  | Inst. Offset  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Remained Inst. | ST  |              Rsvd                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            Inst. List                         ~
|                                                +-+-+-+-+-+-+-+-+
|                                                ~   paddings    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3: The IPv6 Routing Hdr for Instructive Routing

Routing Type Instructive Routing Type

Inst. Offset The offset in the number of octets from the start of Instruction List. The initial value is set to 0 and it points to the 1st instruction to be executed. The value is incremented by the number of octets of the total size of an instruction after the instruction is executed.

Remained Inst. Remained Number of Instructions. The initial value is set to the total number of instructions. The value will be decremented by one after one instruction is executed. The minimum number is one, and it indicates that the end of instruction stack is reached.

ST The satellite address type, default is 0.

Inst. List A list of instructions, the size is variable.

Paddings Pad1 or PadN options to make the packet extension header alignment, see [RFC8200]

6. Instruction List for Instructive Routing

For instructive routing, the instruction list is used to instruct each satellite how to do routing job. The format of the instruction list is illustrated in Figure 4. Each instruction consists of Function Code and Arguments.

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |  Func. Code   |    Arguments  | Func. Code    |  Arguments    ~
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   \--------------\/-------------- \--------------\/--------------/
            instruction[0]                  instruction[1]...
```

Figure 4: The Instruction List for Instructive Routing

Func. Code Function Code, size is 1 octet

Arguments Arguments for the function, Variable length

7. Instructive Routing Behaviors

The behavior for each satellite for instructive routing is described here. Table 1 is the summary of the name, Hex values of all functions, arguments and size. New functions can be defined if needed.

The subsections below are the detailed explanation for each function.

Func Name/Hex Value	Arguments/Size(Octet)	Reference
Fwd.Inc.Sat_ID/0X01	Sat_ID/1	Section 7.1
Fwd.Dec.Sat_ID/0X02	Sat_ID/1	Section 7.2
Fwd.Inc.Obp_ID/0X03	Obp_ID/1	Section 7.3
Fwd.Dec.Obp_ID/0X04	Obp_ID/1	Section 7.4
Fwd.Inc.Shl_ID/0X05	Shl_ID/1	Section 7.5
Fwd.Dec.Shl_ID/0X06	Shl_ID/1	Section 7.6
End.Intf_ID/0X07	Intf_ID/1	Section 7.7
End.Punt/0X08	0X0/1	Section 7.8
End.Lookup/0X09	0X0/1	Section 7.9
End.Lookup.IPv4/0X0A	IPv4_Addr/4	Section 7.10
End.Lookup.IPv6/0X0B	IPv6_Addr/16	Section 7.11
Fwd.Sat_Addr/0X0C	Sat_Addr/4	Section 7.12
Fwd.Sat_MacAddr/0X0D	Sat_MacAddr/6	Section 7.13

Table 1: Functions, Arguments and Reference

The functions in Section 7.1 to Section 7.6 are used for the instructions to forward packet to one of the six major directions discussed in Section 4. They will call API in Section 7.14 to forward the packet to the specified direction.

The functions in Section 7.12 and Section 7.13 are used for the instructions to forward packet to a specified adjacent satellite discussed in Section 4. They will call APIs in Section 7.15 and Section 7.16 respectively to forward the packet to the specified adjacent satellite.

In order to forward packet, each satellite should have an adjacency table stored locally; the table should contain the information about all adjacent satellites, it should at least store:

1. Each adjacent satellite’s semantic address.

2. The ID of local interface connecting to each adjacent satellite.

3. The MAC address for the remote interface of each adjacent satellite.

7.1. Fwd.Inc.Sat_ID

The definition of this function is "Forward the packet on the Satellite Index Incremental Direction until the packet reaches a Satellite whose Satellite Index is equal to the value specified in the argument"

This function is used for the instruction to forward packet to one of the six major directions discussed in Section 4.

When a satellite receives a packet with new routing header, assume the satellite indexes in the address are Shl_index, Obp_index, Sat_index respectively, the satellite does the following. During the forwarding, the Forwarding_API in Section 7.14 is called to forward the packet to the specified direction.

```
S01. When an IRH is processed {
S02.   If ((RI > 1) and (Argument != Sat_index)) {
S03.      Input_Satellite = Current Satellite;
S04.      Input_Direction = Satellite Index Incremental direction;
S05.      Forwarding_API(Packet,Input_Satellite,Input_Direction);
S06.   } else {
S07.      IOF += 2;
S08.      RI --;
S09.      if (RI <= 0)
              Send an ICMP Parameter Problem to the Source Address
              with Code 0 (Erroneous header field encountered)
              and Pointer set to the RI field,
              interrupt packet processing, and discard the packet;
S10.      Proceed to execute the next Instruction;
S11.   }
S12.}
```

7.2. Fwd.Dec.Sat_ID

The definition of this function is "Forward the packet on the Satellite Index Decremental Direction until the packet reaches a Satellite whose Satellite Index is equal to the value specified in the argument"

This function is used for the instruction to forward packet to one of the six major directions discussed in Section 4.

When a satellite receives a packet with new routing header, assume the satellite indexes in the address are Shl_index, Obp_index, Sat_index respectively, the satellite does the following. During the forwarding, the Forwarding_API in Section 7.14 is called to forward the packet to the specified direction.

```
S01. When an IRH is processed {
S02.   If ((RI > 1) and (Argument != Sat_index)) {
S03.      Input_Satellite = Current Satellite;
S04.      Input_Direction = Satellite Index Decremental direction;
S05.      Forwarding_API(Packet,Input_Satellite,Input_Direction);
S06.   } else {
S07.      IOF += 2;
S08.      RI --;
S09.      if (RI <= 0)
              Send an ICMP Parameter Problem to the Source Address
              with Code 0 (Erroneous header field encountered)
              and Pointer set to the RI field,
              interrupt packet processing, and discard the packet;
S10.      Proceed to execute the next Instruction;
S11.   }
S12.}
```

7.3. Fwd.Inc.Opb_ID

The definition of this function is "Forward the packet on the Orbit Plane Index Incremental Direction until the packet reaches a Satellite whose Orbit Plane Index is equal to the value specified in the argument"

This function is used for the instruction to forward packet to one of the six major directions discussed in Section 4.

When a satellite receives a packet with new routing header, assume the satellite indexes in the address are Shl_index, Obp_index, Sat_index respectively, the satellite does the following. During the forwarding, the Forwarding_API in Section 7.14 is called to forward the packet to the specified direction.

```
S01. When an IRH is processed {
S02.   If ((RI > 1) and (Argument != Obp_index)) {
S03.      Input_Satellite = Current Satellite;
S04.      Input_Direction = Orbit Plane Index Incremental direction;
S05.      Forwarding_API(Packet,Input_Satellite,Input_Direction);
S06.   } else {
S07.      IOF += 2;
S08.      RI --;
S09.      if (RI <= 0)
              Send an ICMP Parameter Problem to the Source Address
              with Code 0 (Erroneous header field encountered)
              and Pointer set to the RI field,
              interrupt packet processing, and discard the packet;
S10.      Proceed to execute the next Instruction;
S11.   }
S12.}
```

7.4. Fwd.Dec.Opb_ID

The definition of this function is "Forward the packet on the Orbit Plane Index Decremental Direction until the packet reaches a Satellite whose Orbit Plane Index is equal to the value specified in the argument"

This function is used for the instruction to forward packet to one of the six major directions discussed in Section 4.

When a satellite receives a packet with new routing header, assume the satellite indexes in the address are Shl_index, Obp_index, Sat_index respectively, the satellite does the following. During the forwarding, the Forwarding_API in Section 7.14 is called to forward the packet to the specified direction.

```
S01. When an IRH is processed {
S02.   If ((RI > 1) and (Argument != Obp_index)) {
S03.      Input_Satellite = Current Satellite;
S04.      Input_Direction = Orbit Plane Index Decremental direction;
S05.      Forwarding_API(Packet,Input_Satellite,Input_Direction);
S06.   } else {
S07.      IOF += 2;
S08.      RI --;
S09.      if (RI <= 0)
              Send an ICMP Parameter Problem to the Source Address
              with Code 0 (Erroneous header field encountered)
              and Pointer set to the RI field,
              interrupt packet processing, and discard the packet;
S10.      Proceed to execute the next Instruction;
S11.   }
S12.}
```

7.5. Fwd.Inc.Shl_ID

The definition of this function is "Forward the packet on the Orbit Shell Index Incremental Direction until the packet reaches a Satellite whose Orbit Shell Index is equal to the value specified in the argument"

This function is used for the instruction to forward packet to one of the six major directions discussed in Section 4.

When a satellite receives a packet with new routing header, assume the satellite indexes in the address are Shl_index, Obp_index, Sat_index respectively, the satellite does the following. During the forwarding, the Forwarding_API in Section 7.14 is called to forward the packet to the specified direction.

```
S01. When an IRH is processed {
S02.   If ((RI > 1) and (Argument != Shl_index)) {
S03.      Input_Satellite = Current Satellite;
S04.      Input_Direction = Orbit Shell Index Incremental direction;
S05.      Forwarding_API(Packet,Input_Satellite,Input_Direction);
S06.   } else {
S07.      IOF += 2;
S08.      RI --;
S09.      if (RI <= 0)
              Send an ICMP Parameter Problem to the Source Address
              with Code 0 (Erroneous header field encountered)
              and Pointer set to the RI field,
              interrupt packet processing, and discard the packet;
S10.      Proceed to execute the next Instruction;
S11.   }
S12.}
```

7.6. Fwd.Dec.Shl_ID

The definition of this function is "Forward the packet on the Orbit Shell Index Decremental Direction until the packet reaches a Satellite whose Orbit Shell Index is equal to the value specified in the argument"

This function is used for the instruction to forward packet to one of the six major directions discussed in Section 4.

When a satellite receives a packet with new routing header, assume the satellite indexes in the address are Shl_index, Obp_index, Sat_index respectively, the satellite does the following. During the forwarding, the Forwarding_API in Section 7.14 is called to forward the packet to the specified direction.

```
S01. When an IRH is processed {
S02.   If ((RI > 1) and (Argument != Shl_index)) {
S03.      Input_Satellite = Current Satellite;
S04.      Input_Direction = Orbit Shell Index Decremental direction;
S05.      Forwarding_API(Packet,Input_Satellite,Input_Direction);
S06.   } else {
S07.      IOF += 2;
S08.      RI --;
S09.      if (RI <= 0)
              Send an ICMP Parameter Problem to the Source Address
              with Code 0 (Erroneous header field encountered)
              and Pointer set to the RI field,
              interrupt packet processing, and discard the packet;
S10.      Proceed to execute the next Instruction;
S11.   }
S12.}
```

7.7. End.Intf_ID

The definition of this function is "End of processing for the Instructive routing, remove the Instructive Routing Header, Forward the packet to the interface specified in the argument"

This function is normally used on the Dst_Sat to forward packet to Dst_GS.

When a satellite receives a packet with new routing header, the satellite does the following, Forwarding_GS_API in Section 7.17 is called to forward the packet to the specified interface.

```
S01. When an IRH is processed {
S02.   Change the Next header in the packet header to be
          the Next Header field in the Instructive Routing header;
S03.   Remove the Instructive Routing Header;
S04.   Forwarding_GS_API(Packet, Argument);
S05.}
```

7.8. End.Punt

The definition of this function is "End of processing for the Instructive routing, remove the Instructive Routing Header, Punt the packet to the OS for process"

This function is normally used send packet to a satellite. At the destination satellite, the packet is punted to the OS to be processed further.

When a satellite receives a packet with new routing header, the satellite does the following:

```
S01. When an IRH is processed {
S02.   Change the Next header in the packet header to be
         the Next Header field in the Instructive Routing header;
S03.   Remove the Instructive Routing Header;
S04.   Punt packet to the local CPU for process;
S05.}
```

7.9. End.Lookup

The definition of this function is "End of processing for the Instructive routing, remove the Instructive Routing Header, Lookup the destination address in packet header and forward the packet accordingly"

This function is normally used to send packet to Dst_GS. After the packet reaches the Dst_Sat, the packet is forwarded to Dst_GS by looking up the destination address in the IPv6 packet header.

When a satellite receives a packet with new routing header, the satellite does the following:

```
S01. When an IRH is processed {
S02.   Change the Next header in the packet header to be
         the Next Header field in the Instructive Routing header;
S03.   Remove the Instructive Routing Header;
S04.   Lookup the destination address in packet hdr and forward
         the packet;
S05.}
```

7.10. End.Lookup.IPv4

The definition of this function is "End of processing for the Instructive routing, remove the Instructive Routing Header, Lookup the IPv4 address specified in the argument and forward the packet accordingly"

This function is normally used to send packet to Dst_GS. After the packet reaches the Dst_Sat, the packet is forwarded to Dst_GS by looking up the IPv4 destination address specified in the Function Argument.

When a satellite receives a packet with new routing header, the satellite does the following:

```
S01. When an IRH is processed {
S02.   Fetch the IPv4 addr in the argument;
S03.   Change the Next header in the packet header to be
         the Next Header field in the Instructive Routing header;
S04.   Remove the Instructive Routing Header;
S05.   Lookup the fetched IPv4 address and forward the packet;
S06.}
```

7.11. End.Lookup.IPv6

The definition of this function is "End of processing for the Instructive routing, remove the Instructive Routing Header, Lookup the IPv6 address specified in the argument and forward the packet accordingly"

This function is normally used to send packet to Dst_GS. After the packet reaches the Dst_Sat, the packet is forwarded to Dst_GS by looking up the IPv6 destination address specified in the Function Argument.

When a satellite receives a packet with new routing header, the satellite does the following:

```
S01. When an IRH is processed {
S02.   Fetch the IPv6 addr in the argument;
S03.   Change the Next header in the packet header to be
         the Next Header field in the Instructive Routing header;
S04.   Remove the Instructive Routing Header;
S05.   Lookup the fetched IPv6 address and forward the packet;
S06.}
```

7.12. Fwd.Sat_Addr

The definition of this function is "Forward the packet to the adjacent satellite with the address specified in the argument"

This function is normally used for the instruction to forward packet to an adjacent satellite specified by its Satellite Semantic Address. The Satellite Semantic Address is 32-bit long and is defined in Section 5.4 in [I-D.lhan-satellite-semantic-addressing]

When a satellite receives a packet with new routing header, assume the satellite semantic address is Sat_Addr, the satellite does the following:

```
S01. When an IRH is processed {
S02.   If ((RI > 1) and (Argument != Sat_Addr)) {
S03.      Input_Satellite = Current Satellite;
S04.      SatAddr = Argument;
S05.      Forwarding_API_SAT(Packet,Input_Satellite,SatAddr);
S06.   } else {
S07.      IOF += 4;
S08.      RI --;
S09.      if (RI <= 0)
              Send an ICMP Parameter Problem to the Source Address
              with Code 0 (Erroneous header field encountered)
              and Pointer set to the RI field,
              interrupt packet processing, and discard the packet.
S10.      Proceed to execute the next Instruction;
S11.   }
S12.}
```

7.13. Fwd.Sat_MacAddr

The definition of this function is "Forward the packet to the adjacent satellite with the MAC address specified as the argument"

This function is normally used for the instruction to forward packet to an adjacent satellite specified by its MAC address.

When a satellite receives a packet with new routing header, assume the satellite Mac address is Sat_MacAddr, the satellite does the following:

```
S01. When an IRH is processed {
S02.   If ((RI > 1) and (Argument != Sat_MacAddr)) {
S03.      Input_Satellite = Current Satellite;
S04.      SatMacAddr = Argument;
S05.      Forwarding_API_Mac(Packet,Input_Satellite,SatMacAddr);
S06.   } else {
S07.      IOF += 6;
S08.      RI --;
S09.      if (RI <= 0)
              Send an ICMP Parameter Problem to the Source Address
              with Code 0 (Erroneous header field encountered)
              and Pointer set to the RI field,
              interrupt packet processing, and discard the packet.
S10.      Proceed to execute the next Instruction;
S11.   }
S12.}
```

7.14. Forwarding_API(Packet,Input_Satellite,Input_Direction)

This API will forward a packet to the specified direction. When a satellite executes the API, it will do following:

```
S01. Forwarding_API(Packet,Input_Satellite,Input_Direction) {
S02.    Lookup the local adjacency table to find out
           1) The adjacent satellite of "Input_Satellite" on the
              direction equal to "Input_Direction" (The adjacent
              satellite's semantic address can be inferred by
              the "Input_Satellite" and "Input_Direction").
           2) The L2 address for the adjacent satellite;
           3) The local interface connecting to the adjacent
              satellite;
S03.    Rewrite the L2 header of the Packet by the L2 address;
S04.    Send the Packet to the local interface;
S05.}
```

7.15. Forwarding_API_SAT(Packet,Input_Satellite,Sat_Addr)

This API will forward a packet to the specified adjacent satellite with the semantic address as the argument. When a satellite executes the API, it will do following:

```
S01. Forwarding_API_SAT(Packet,Input_Satellite,SatAddr) {
S02.    Lookup the local adjacency table to find out
           1) The adjacent satellite of "Input_Satellite"
              (The adjacent satellite address is SatAddr);
           2) The L2 address for the adjacent satellite;
           3) The local interface connecting to the adjacent
              satellite;
S03.    Rewrite the L2 header of the Packet by the L2 address;
S04.    Send the Packet to the local interface;
S05.}
```

7.16. Forwarding_API_MAC(Packet,Input_Satellite,Sat_MacAddr)

This API will forward a packet to the specified adjacent satellite with the MAC address as the argument. When a satellite executes the API, it will do following:

```
S01. Forwarding_API_MAC(Packet,Input_Satellite,SatMacAddr) {
S02.    Lookup the local adjacency table to find out
           1) The adjacent satellite of "Input_Satellite"
              (The adjacent satellite MAC address is SatMacAddr);
           2) The L2 address for the adjacent satellite;
           3) The local interface connecting to the adjacent
              satellite;
S03.    Rewrite the L2 header of the Packet by the L2 address;
S04.    Send the Packet to the local interface;
S05.}
```

7.17. Forwarding_GS_API(Packet,Input_Interface)

This API will forward a packet to ground station the connected to the specified interface. When a satellite executes the API, it will do following:

```
S01. Forwarding_API(Packet,Input_Interface) {
S02.    Lookup the local adjacency table to find out
           1) The connected GS to the interface
              equal to "Input_Interface";
           2) The L2 address for the GS;
S03.    Rewrite the L2 header of the Packet by the L2 address;
S04.    Send the Packet to the "Input_Interface";
S05.}
```

8. Other notes

Due to the limit of the picture drawing for IETF draft, the pictures in the memo may not be easy to understand. For easier understanding of the method, please refere to the [Large-Scale-LEO-Network-Routing], it provided more vivid pictures obtained by simulation software Savi [Savi], and also provided the simulation results.

9. IANA Considerations

This document defines a new IPv6 Routing Type: the "Instructive Routing Header". It needs to be assigned a number by IANA.

This document also defines an 8-bit Function Name, for which IANA will create and will maintain a new sub-registry entitled "Instructive Routing Function Name" under the "Internet Protocol Version 6 (IPv6) Parameters" [IPv6_Parameters] registry. Initial values for the subtype registries are given in Table 1.

10. Security Considerations

The instructive routing is only applicable to a satellite network that is using the satellite semantic address. It will add instructive routing header at a GS and the header will be removed before reaching another GS. Normally, a satellite network including all GS is trusted domain. Traffic will be filtered at the domain boundaries. Non-authorized users cannot access the satellite network.

11. Contributors

12. Acknowledgements

13. References

13.1. Normative References

[RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <https://www.rfc-editor.org/info/rfc8200>.

[RFC8754] Filsfils, C., Ed., Dukes, D., Ed., Previdi, S., Leddy, J., Matsushima, S., and D. Voyer, "IPv6 Segment Routing Header (SRH)", RFC 8754, DOI 10.17487/RFC8754, March 2020, <https://www.rfc-editor.org/info/rfc8754>.

[RFC8986] Filsfils, C., Ed., Camarillo, P., Ed., Leddy, J., Voyer, D., Matsushima, S., and Z. Li, "Segment Routing over IPv6 (SRv6) Network Programming", RFC 8986, DOI 10.17487/RFC8986, February 2021, <https://www.rfc-editor.org/info/rfc8986>.

13.2. Informative References

[I-D.lhan-problems-requirements-satellite-net] Han, L., Li, R., Retana, A., Chen, M., Su, L., Jiang, T., and N. Wang, "Problems and Requirements of Satellite Constellation for Internet", Work in Progress, Internet-Draft, draft-lhan-problems-requirements-satellite-net-05, 5 July 2023, <https://datatracker.ietf.org/doc/html/draft-lhan-problems-requirements-satellite-net-05>.

[I-D.lhan-satellite-semantic-addressing] Han, L., Li, R., Retana, A., Chen, M., and N. Wang, "Satellite Semantic Addressing for Satellite

Constellation", Work in Progress, Internet-Draft, draft-lhan-satellite-semantic-addressing-04, 1 September 2023, <https://datatracker.ietf.org/doc/html/draft-lhan-satellite-semantic-addressing-04>.

[I-D.retana-lsr-ospf-monitor-node]
Retana, A. and L. Han, "OSPF Monitor Node", Work in Progress, Internet-Draft, draft-retana-lsr-ospf-monitor-node-00, 7 March 2022, <https://datatracker.ietf.org/doc/html/draft-retana-lsr-ospf-monitor-node-00>.

[StarLink] "Star Link", <https://en.wikipedia.org/wiki/Starlink>.

[ThreeGPP] "3GPP", <https://www.3gpp.org/>.

[OSI-Model]
"OSI Model", <https://en.wikipedia.org/wiki/OSI_model>.

[TR38-821] "Solutions for NR to support Non-Terrestrial Networks (NTN)", <https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3525>.

[TR23-700] "Study on support of satellite backhauling in 5GS", <https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=4000>.

[IPv6_Parameters]
IANA, "Internet Protocol Version 6 (IPv6) Parameters", <https://www.iana.org/assignments/ipv6-parameters/ipv6-parameters.xhtml#ipv6-parameters-3>.

[Large-Scale-LEO-Network-Routing]
Han, L.,Retana, A., Westphal, C. and R. Li, "Large Scale LEO Satellite Networks for the Future Internet: Challenges and Solutions to Addressing and Routing," Computer Networks and Communications, 1(1), 31-58", 2023, <https://ojs.wiserpub.com/index.php/CNC/article/view/2105>.

[Savi] "Satellite constellation visualization", <https://savi.sourceforge.io/>.

Appendix A. Change Log

* Initial version, 02/28/2022

* Revision 1, 09/02/2022

* Revision 2, 03/03/2023

* Revision 3, 09/01/2023

Authors' Addresses

Lin Han (editor)
Futurewei Technologies, Inc.
2330 Central Express Way
Santa Clara, CA 95050,
United States of America
Email: lhan@futurewei.com

Alvaro Retana
Futurewei Technologies, Inc.
2330 Central Express Way
Santa Clara, CA 95050,
United States of America
Email: alvaro.retana@futurewei.com

Richard Li
Futurewei Technologies, Inc.
2330 Central Express Way
Santa Clara, CA 95050,
United States of America
Email: rli@futurewei.com

Network Working Group
Internet-Draft
Intended status: Informational
Expires: 4 March 2024

L. Han, Ed.
R. Li
A. Retana
Futurewei Technologies, Inc.
M. Chen
China Mobile
N. Wang
University of Surrey
1 September 2023

Satellite Semantic Addressing for Satellite Constellation
draft-lhan-satellite-semantic-addressing-04

Abstract

This document presents a semantic addressing method for satellites in satellite constellation connecting with Internet. The satellite semantic address can indicate the relative position of satellites in a constellation. The address can be used with traditional IP address or MAC address or used independently for IP routing and switching.

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 4 March 2024.

Table of Contents

1. Introduction

Satellite constellation technologies for Internet are emerging and expected to provide Internet service like the traditional wired network on the ground. A typical satellite constellation will have couple of thousands or over ten thousand of LEO and/or VLEO. Satellites in a constellation will be connected to adjacent satellites by Inter-Satellite-Links (ISL), and/or connected to ground station by microwave or laser links. ISL is still in research stage and will be deployed soon. This memo is for the satellite networking with the use of ISL.

The memo proposes to use some indexes to represent a satellite's orbit information. The indexes can form satellite semantic address, the address can then be embedded into IPv6 address or MAC address for IP routing and switching. The address can also be used independently if the shorter than 128-bit length of IP address is accepted. As an

internal address for satellite network, it only applies to satellites that will form a constellation to transport Internet traffic between ground stations and will not be populated to Internet by BGP.

2. Terminology

LEO	Low Earth Orbit with the altitude from 180 km to 2000 km.
VLEO	Very Low Earth Orbit with the altitude below 450 km
GEO	Geosynchronous orbit with the altitude 35786 km
ISL	Inter Satellite Link
ISLL	Inter Satellite Laser Link
3D	Three Dimensional
GS	Ground Station, a device on ground connecting the satellite. In the document, GS will hypothetically provide L2 and/or L3 functionality in addition to process/send/receive radio wave. It might be different as the reality that the device to process/send/receive radio wave and the device to provide L2 and/or L3 functionality could be separated.
SGS	Source ground station. For a specified flow, a ground station that will send data to a satellite through its uplink.
DGS	Destination ground station. For a specified flow, a ground station that is connected to a local network or Internet, it will receive data from a satellite through its downlink and then forward to a local network or Internet.
L1	Layer 1, or Physical Layer in OSI model [OSI-Model]
L2	Layer 2, or Data Link Layer in OSI model [OSI-Model]
L3	Layer 3, or Network Layer in OSI model [OSI-Model], it is also called IP layer in TCP/IP model
BGP	Border Gateway Protocol [RFC4271]

IGP Interior gateway protocol, examples of IGPs include Open Shortest Path First (OSPF [RFC2328]), Routing Information Protocol (RIP [RFC2453]), Intermediate System to Intermediate System (IS-IS [RFC7142]) and Enhanced Interior Gateway Routing Protocol (EIGRP [RFC7868]).

3. Overview

For IP based satellite networking, the topology is very dynamic and the traditional IGP and BGP based routing technologies will face challenges according to the analysis in [I-D.lhan-problems-requirements-satellite-net]. From the paper, we can easily categorize satellite links as two types, steady and un-steady. For un-steady links, the link status will be flipping every couple of minutes.

Section 5.5 has more details about how to identify different links.

Some researches have been done to handle such fast changed topologies. one method to overcome the difficulties for routing with un-steady links is to only use the steady links, and get rid of un-steady links unless it is necessary. For example, for real deployment, only links between satellite and ground stations are mandatory to use, other un-steady links can be avoided in routing and switching algorithms. [Routing-for-LEO] proposed to calculate the shortest path by avoiding un-steady links in polar area and links crossing Seam line since satellites will move in the opposite direction crossing the Seam line.

Traditionally, to establish an IP network for satellites, each satellite and its interface between satellites and to ground stations have to be assigned IP addresses (IPv4 or IPv6). The IP address can be either private or public. IP address itself does not mean anything except routing prefix and interface identifier [RFC8200].

To utilize the satellite relative position for routing, it is desired that there is an easy way to identify the relative positions of different satellites and identify un-steady links quickly. The traditional IP address cannot provide such functionality unless we have the real-time processing for 3D coordinates of satellites to figure out the relative positions of each satellite, and some math calculation and dynamic database are also needed in routing algorithm to check if a link is steady or not. This will introduce extra data exchanged for routing protocols and burden for the computation in every satellite. Considering the ISL link speed (up to 10G for 2000km) and hardware cost (Radiation-hardened semiconductor components are needed) in satellite are more constraint than for network device on ground, it is expected to simplify the routing algorithm, reduce the requirement of ISL, onboard CPU and memory.

The document proposes to form a semantic address by satellite orbit information, and then embedded it into a proper IP address. The IP address of IGP neighbors can directly tell the relative position of different satellites and if links between two satellites are stead or not.

The document does not describe the details how the semantic address is used to improve routing and switching or new routing protocols, those will be addressed in different documents. Instructive routing [I-D.lhan-satellite-instructive-routing] is a new proposal to use the semantic address for the routing of large-scale LEO satellite network. It is based on source routing mechnism and meshing characteristics of LEO satellite constellation, using semantic address can reduce the overheader of the instruction for the packet forwarding at each satellite. The complete solution combining the semantic address, the instructive routing and modified OSPF [I-D.retana-lsr-ospf-monitor-node] can be found in [Large-Scale-LEO-Network-Routing].

4. Basics of Satellite Constellation and Satellite Orbit

This section will introduce some basics for satellite such as orbit parameters.

4.1. Satellite Orbit

The orbit of a satellite can be either circular or ecliptic, it can be described by following Keplerian elements [KeplerianElement]:

1. Inclination (i)

2. Longitude of the ascending node (Omega)

3. Eccentricity (e)

4. Semimajor axis (a)

5. Argument of periapsis (omega)

6. True anomaly (nu)

The circular orbit is widely used by proposals of satellite constellation from different companies and countries.

For a circular orbit, we will have:

* Eccentricity e = 0

* Semimajor axis a = Altitude of satellite

* Argument of periapsis omega = 90 degree

So, three parameters, Altitude, Inclination and Longitude of the ascending node, will be enough to describe the orbit. The satellite will move in a constant speed and True anomaly (nu) can be easily calculated after the epoch time is defined.

4.2. Satellite Constellation Compositions

One satellite constellation may be composed of many satellites (LEO and VLEO), but normally all satellites are grouped in a certain order that is never changed during the life of satellite constellation. Each satellite constellation's orbits parameters described in Section 4.1 must be approved by regulator and cannot be changed either. Follows are characters of one satellite constellation:

1. One Satellite Constellation is composed of couple of shell groups of satellites.

2. The same shell group of satellites will have the same altitude and inclination angle.

3. The total No orbit planes in the same shell group of satellites will be evenly distributed by the same interval of Longitude of the ascending node (Omega). The interval equals to (360 degree/ No). As a result, all orbit planes in the same shell group will effectively form a shell to cover earth (there will be a coverage hole for the shell on the sky in both polar areas if the inclination angle is less than 90 degree).

4. Each orbit plane in the same shell group will have the same number of satellites, all satellites in the same orbit plane will be evenly distributed angularly in the orbit plane. Assuming there are Ns satellites in each orbit plane, then the angular interval of satellites equals to (360 degree/Ns).

5. All satellites in the same shell group are moving in the same circular direction. As a result, at any location on earth, we can see there will have two group of satellites moving on the opposite direction. One group moves from south to north, and another group moves from north to south. Section 5.5 has more details.

4.3. Communication between Satellites by ISL

When ISL is used for the communication between satellites, each satellite will have a fixed number of links to connect to its neighbor. Due to the cost of ISL and the constraints of power supply on satellite, the number of ISL is normally limited to connect to its closest neighbors. In 3D space, each satellite may have six types of adjacent satellites, each type represents one direction. The number of adjacent neighbors in one direction is dependent on the number of deployment of ISL device on satellites, for example, the laser transmitter and receiver for ISLL. Figure 1 illustrates satellite S0 and its adjacent neighbors.

```
                /           /           /
               /           /           /
              /           /           /
             S7          S8          S9
            /           /           /
           /           /           /
          /           /           /
                /           S1          /
               S5          /           S3
              /           /           /
             /           S0          /
            /           /           /
           S6          /           S4
          /           S2          /
                /           /           /
               /           /           /
              /           /           /
             S10         S11         S12
            /           /           /         ^ Moving direction
           /           /           /         /
          /           /           /         /
         orbit       orbit       orbit
```

Figure 1: Satellite S0 and its adjacent neighbors

All adjacent satellites of S0 in Figure 1 are listed below:

1. The front adjacent satellite S1 that is on the same orbit plane as S0.

2. The back adjacent satellite S2 that is on the same orbit plane as S0

3. The right adjacent satellites S3 and S4 that are on the right orbit plane of S0

4. The left adjacent satellites S5 and S6 that are on the left orbit plane of S0

5. The above adjacent satellites S7 to S9 that are on the above orbit plane of S0

6. The below adjacent satellite S10 to S12 that are on the below orbit of plane S0

The relative position of adjacent satellites will directly determine the quality of ISL and communication. From the analysis in [I-D.lhan-problems-requirements-satellite-net], The speed of satellite is only related to the altitude of the satellite (on circular orbit), all satellites with a same altitude will move with the same speed. So, in above adjacent satellites, some adjacent satellite's relative positions are steady and the ISL can be alive without interruption caused by movement. Some adjacent satellites relative positions are changing quickly, the ISL may be down since the distance may become out of reach for the laser of ISL, or the quick changed positions of two satellite make the tracking of laser too hard. Below are details:

* The relative position of satellites in the same orbit plane will be the steadiest.

* The relative position of satellites in the direct neighbor orbit planes in the same shell group and moving in the same direction will be steady at equator area, but will be changing when two orbits meet on the polar area. Whether the link status will be flipping depends on the tracking technology and the range of laser pointing angle of ISL. See Figure 2.

* The relative position of satellites in the neighbor orbit planes in the same shell group but moving in the different direction will not be steady at all times. More details are explained in Figure 8

* The relative position of satellites in the neighbor orbit planes in the different shell group will be dependent on the difference of altitude and inclination. This has been analyzed in [I-D.lhan-problems-requirements-satellite-net].

```
\      /
 P3   P4
  \  /
   \/
   /\
  /  \
 P1   P2
/      \
```

* Two satellites S1 and S2 are at position P1 and P2 at time T1
* S1's right facing ISL connected to S2's left facing ISL
* S1 and S2 move to the position P4 and P3 at time T2
* S1's left facing ISL connected to S2's right facing ISL
* So, if the range of laser pointing angle is 360 degree and tracking technology supports, the ISL will not be flipping after passing polar area; Otherwise, the link will be flipping

Figure 2: Satellite's Position and ISL Change at Polar Area

5. Addressing of Satellite

When ISL is deployed in satellite constellation, all satellites in the constellation can form a network like the wired network on ground. Due to the big number of satellites in a constellation, the network could be either L2 or L3. The document proposes to use L3 network for better scalability.

When satellites form a L3 network, it is expected that IP address is needed for each satellite and its ISLs.

While the traditional IP address can still be used for satellite network, the document proposes an alternative new method for satellite's addressing system. The new addressing system can indicate a satellite's orbit info such as shell group index, orbit plane index and satellite index. This will make the adjacent satellite identification for link status easier and benefit the routing algorithms.

5.1. Indexes of Satellite

As described in Section 4.2, one satellite has three important orbit related information as described below.

1. Index for the shell group of satellites in a satellite constellation

2. Index for the orbit plane in a shell group of satellites

3. Index for the satellite in an orbit plane

It should be noted that for all type of indexes, it is up to the owner to assign the index number. There is no rule for which one should be assigned with which number. The only important rule is that all index number should be in sequential to reflect its relative order and position with others. Below is an example of assignment rules:

1. The 1st satellite launched in an orbit plane can be assigned for the 1st satellite index (0), the incremental direction of the satellite index in the same orbit plane is the incremental direction of "Argument of periapsis (omega)"

2. The 1st orbit plane established can be assigned for the 1st orbit plane index (0), the incremental direction of the orbit plane index is the incremental direction of "Longitude of the ascending node (Omega)".

3. The shell group of satellites with the lowest altitude can be assigned for the 1st shell group index (0), the incremental direction of shell group index is the incremental direction of altitude.

It should also be noted that for all type of indexes assignment, there are no strict requirement for the physical positions of satellite. Due to the launching time difference, the shifing of the satellite orbit after some time, the orbit parameters of satellites always have some difference and do not follow the theoritical values. For example:

1. The altitude of all satellites in the same shell group might not be exactly same.

2. The inclination angle of all satellites in the same shell group might not be exactly same.

3. The Longitude of the ascending node (Omega) of all satellites in the same orbit plane might not be exactly same.

4. The interval of the Longitude of the ascending node (Omega) of all orbit plane in the same shell group might not be equal

5. The angular interval of all satellites in the same orbit plane might not be equal.

Figure 3 and Figure 4 illustrate three types of indexes for satellite

```
           /           /           /   \
          /           /           /    |
         /           /           /     |
        S           S           S      > shell group3
       /           /           /       |
      /           /           /        |
     /           /           /         /
           /           S           /   \
          S           /           S    |
         /           /           /     |
        /           S           /      > shell group2
       /           /           /       |
      S           /           S        |
     /           S           /         /
           /           /           /   \
          /           /           /    |
         /           /           /     |
        S           S           S      > shell group1
       /           /           /       |
      /           /           /        |
     /           /           /         /
   orbit     orbit      orbit           ----> Earth self-rotation
   plane1    plane2     plane3
```

Figure 3: Shell Group and Orbit Plane Indexes for Satellites

Shell Group and Orbit Plane Indexes for Satellites

```
                 , - ~ S1 ~ - ,
             S2 ,               , S8
           ,                       ,
          ,                         ,
         ,                           ,      Indexed
         S3                          S7 <-- satellite
         ,                           ,      in one orbit plane
          ,                         ,
           ,                       ,      ^  move direction
             S4                  , S6     /
               ' - , _ S5_ ,  '          /
```

Figure 4

Three types of Index for satellites

5.2. The Range of Satellite Indexes

The ranges of different satellite indexes will determine the range the dedicated field for semantic address. The maximum indexes depend on the number of shell group, orbit plane and satellite per orbit plane. The number of orbit plane and satellite per orbit plane have relationship with the coverage of a satellite constellation. There are minimum numbers required to cover earth. [I-D.lhan-problems-requirements-satellite-net] has given the detailed math to estimate the minimal number required to cover the earth. There are two key parameters that determine the minimal number of satellite required. One is the elevation angle, another is the altitude. StarLink has proposed two elevation angles, 25 and 35 degrees [SpaceX-Non-GEO]. The lowest LEO altitude can be 160km according to [Lowest-LEO-ESA]. The Table 1 and Table 2 illustrate the estimation for different altitude (As), the coverage radius (Rc), the minimal required number of orbit planes (No) and satellite per orbit plane (Ns). The elevation angle is 25 degree and 35 degrees respectively.

Parameters	VLEO1	VLEO2	LEO1	LEO2	LEO3	LEO4	LEO5
As(km)	160	300	600	900	1200	1500	2000
Rc(km)	318	562	1009	1382	1702	1981	2379
Ns	73	42	23	17	14	12	10
No	85	48	27	20	16	14	12

Table 1: Satellite coverage (Rc), minimal number of orbit plane (No) and satellite (Ns) per orbit plane for different LEO/VLEOs, Elevation angle = 25 degree

Parameters	VLEO1	VLEO2	LEO1	LEO2	LEO3	LEO4	LEO5
As(km)	160	300	600	900	1200	1500	2000
Rc(km)	218	392	726	1015	1271	1498	1828
Ns	107	59	32	23	19	16	13
No	123	69	37	27	22	18	15

Table 2: Satellite coverage (Rc), minimal number of orbit plane (No) and satellite (Ns) per orbit for different LEO/ VLEOs, Elevation angle = 35 degree

The real deployment may be different as above analysis. Normally, more satellites and orbit planes are used to provide better coverage. So far, there are only two proposals available, one is StarLink, another is from China Constellation. For proposals of [StarLink], there are 7 shell groups, the number of orbit plane and satellites per orbit plane in all shell groups are 72 and 58; For proposals of [China-constellation], there are 7 shell groups, the number of orbit plane and satellites per orbit plane in all shell groups are 60 and 60;

It should be noted that some technical parameters, such as the inclination and altitude of orbit planes, in above proposals may be changed during the long-time deployment period, but the total numbers for indexes normally do not change.

From the above analysis, to be conservative, it is safe to conclude that the range of all three satellite indexes are less than 256, or 8-bit number.

5.3. Other Info for satellite addressing

In addition to three satellite indexes described in Section 5.1, other information is also important and can also be embedded into satellite address:

1. The company or country code, or the owner code. In the future, there may have multiple satellite constellations on the sky from different organizations, and the inter-constellation communication may become as normal that is similar to the network on the ground. This code will be useful to distinguish different satellite constellation and make the inter-constellation communication possible. One satellite constellation will have one code assigned by international regulator (IANA or ITU). Considering the following facts:

   * The space of LEO satellite orbits is limited. New LEO satellite orbits need ITU's approve.

   * The spectrum for LEO satellite communication is limited. New spectrum needs ITU's approve.

   * The costs of satellite constellations in launching, maintenance and operation are considerably high.

   We can predict the total number of satellite constellation is very limited. So, the size of code is limited. In the draft, we propose to use one octet for Owner code.

2. The Interface Index. This index is to identify the ISL or ISLL for a satellite. As described in Section 4.3, the total number of ISL is limited. So, the size of interface index is also limited.

5.4. Encoding of Satellite Semantic Address

The encoding for satellite semantic address is dependent on what routing and switching (L2 or L3 solution) technologies are used for satellite networking, and finally dependent on the decision of IETF community.

Follows are some initial proposals:

1. 32-bit satellite semantic address (Figure 5) can be used for Router ID if IGP, i.e, OSPF, is used for the routing within the satellite network. Note, this does not hint the current OSPF can be used for satellite network without any changes. Separate drafts should be written to describe the details about the modified OSPF for satellite network routing.

2. When satellite network is using L3 or IPv6 solution, the satellite semantic address is encoded as the interface identifier (i.e., the rightmost 64 bits) of the IPv6 address for IPv6. Figure 6 shows the format of IPv6 Satellite Address.

3. When satellite network is using L2 solution, the satellite semantic address can be embedded into the field of "Network Interface Controller (NIC) Specific" in MAC address [IEEE-MAC-Address]. But due to shorter space for NIC, the "Index for the shell group" and "Index for Interface" will only have 4-bit. This is illustrated in Figure 7. This encoded MAC address can also be used for L3 solution where the interface MAC may be also needed to be configured for each ISL.

4. Recently, some works suggested to use Length Variable IP address for routing and switching [Length-Variable-IP] or use flexible IP address [I-D.jia-flex-ip-address-structure] or shorter IP address [I-D.li-native-short-addresses] to solve some specific problems that regular IPv6 is not very suitable. Satellite network also belongs to such specific network. Due to the resource and cost constraints and requirement for radiation hardened electronic components, the ISL speed, on-board processor and memory are limited in performance, power consumption and capacity compared with network devices on ground. So, using IPv6 directly in satellite network is not an optimal solution because IPv6 header size is too long for such small network. From above analysis, 32-bit to 64-bit length of IP address is enough for satellite networking. Using 128-bit IPv6 will consume more resource especially the ISL bandwidth, processing power and memory, etc. If shorter than 128-bit IP address is accepted as IETF work, the satellite semantic address can be categorized as a similar use case. Figure 5 illustrates a 32-bit Semantic Satellite Address format. The final coding for the shorter IP address can be decided by the community. How to use the 32-bit Semantic Satellite address can be addressed later on in different document.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Owner Code  |  Shell_Index  |  Orbit_Index  |   Sat_Index   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Owner Code: Identifier for the owner of the constellation
Shell_Index: Index for the shell group of satellite in a satellite constellation
Orbit_Index: Index for the orbit plane in a shell group of satellite
Sat_Index: Index for the satellite in an orbit plane

Figure 5: The 32-bit Semantic Satellite Address

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                     Subnet Prefix (64 bits)                   ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Owner Code  |  Shell_Index  |  Orbit_Index  |   Sat_Index   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Intf_Index  |                    Reserved                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Owner Code: Identifier for the owner of the constellation
Shell_Index: Index for the shell group of satellite in a satellite constellation
Orbit_Index: Index for the orbit plane in a shell group of satellite
Sat_Index: Index for the satellite in an orbit plane
Intf_Index: Index for interface on a satellite
Reserved: 24-bits reserved

Figure 6: The IPv6 Satellite Address

```
              3 Octets             3 Octets
        /---------^--------\ /--------^--------\
        +-------------------+-------------------+
        |        OUI        |     Sat Address   |
        +-------------------+-------------------+
                                     |
                                     |
     +-------------------------------+
     |
     |
     v

      0                   1                   2                   3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     | Shell |  Orbit_Index  |   Sat_Index   |Intf_Id|
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

OUI: Organizationally Unique Identifier assigned by IEEE
Shell: 4-bit Index for the shell group of satellite in a satellite constellation
Orbit_Index: Index for the orbit plane in the group of satellite
Sat_Index: Index for the satellite in the orbit plane
Intf_Id: 4-bit Index for interface on a satellite

Figure 7: The MAC Satellite Address

5.5. Link Identification by Satellite Semantic Address

Using above satellite semantic addressing scheme, to identify steady and un-steady links is as simple as below:

Assuming:

1. The total number of satellites per orbit plane is M

2. The total number of orbit planes per shell group is N.

3. Two satellites have:

   * Satellite Indexes as: Sat1_Index, Sat2_Index

   * Orbit plane Indexes as: Orbit1_Index, Orbit2_Index

   * Shell group Indexes as: Shell1_Index, Shell2_Index

Steady links:

1. The links between adjacent satellites on the same orbit plane, or, the satellite indexes satisfy:

   * Sat2_Index = Sat1_Index + 1, when Sat1_Index < M-1; Sat2_Index = 0, when Sat1_Index = M-1; and

   * Orbit1_Index = Orbit2_Index, Shell1_Index = Shell2_Index.

2. The links between satellites on adjacent orbit planes on the same altitude. and two satellites are moving to the same direction, or, the satellite indexes satisfy:

   * Orbit2_Index = Orbit1_Index + 1, when Orbit1_Index < N-1; Orbit2_Index = 0, when Orbit1_Index = N-1; and

   * Shell1_Index = Shell2_Index.

   * Sat1_Index and Sat2_Index may be equal or have difference, depend on how the link is established.

Un-Steady links:

1. The links between satellite and ground stations.

2. The links between satellites on adjacent orbit planes on the same altitude. Two satellites are moving to the different direction. Or, the satellite indexes do not satisfy conditions described in above #2 for Steady links.

3. The links between satellites on adjacent orbit planes on different altitude. Or, the satellite indexes satisfy:

   * Shell1_Index != Shell2_Index.

Figure 8 illustrates the links for adjacent orbit planes (#2 for Steady Link and Un-steady Link above). From the figure, it can be noticed that some links may have shorter distance than steady link, but they are unsteady. For example, the links between S1 and S4; S4 and S2; S2 and S5, etc.

```
                 i+N/2         i+1+N/2       i+2+N/2
               / \           / \           / \
              /   \         /   \         /   \
             S1............S2............S3    \
            /       S4 ..........S5............S6
           /         \   /         \   /         \
          /           \ /           \ /           \
          i-1           i             i+1
```

* The total number of orbit planes are N
* The number (i-1, i, i+1,...) represents the Orbit index
* The bottom numbers (i-1, i, i+1) are for orbit planes on which satellites (S1, S2, S3) are moving from bottom to up.
* The top numbers (i+N/2, i+1+N/2, i+2+N/2) are for orbit planes on which satellites (S4, S5, S6) are moving from up to bottom.
* Dot lines are the steady links

Figure 8: The links between satellites on adjacent orbit planes

6. Other notes

Due to the limit of the picture drawing for IETF draft, the pictures in the memo may not be easy to understand. For easier understanding of the method, please refere to the [Large-Scale-LEO-Network-Routing], it provided more vivid pictures obtained by simulation software Savi [Savi].

7. IANA Considerations

This memo may include request to IANA for owner code, see Section 5.4.

8. Security Considerations

The semantic address for satellite only describes the relative positions of satellites, it does not introduce more security issues compared with the normal IP address. Similar to terrestrial network, a satellite network normally will have different protocols at the different layers, form L1 to L7, to provide the security for a satellite network.

9. Contributors

10. Acknowledgements

11. References

11.1. Normative References

[RFC4271] Rekhter, Y., Ed., Li, T., Ed., and S. Hares, Ed., "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, DOI 10.17487/RFC4271, January 2006, <https://www.rfc-editor.org/info/rfc4271>.

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

[RFC7142] Shand, M. and L. Ginsberg, "Reclassification of RFC 1142 to Historic", RFC 7142, DOI 10.17487/RFC7142, February 2014, <https://www.rfc-editor.org/info/rfc7142>.

[RFC2453] Malkin, G., "RIP Version 2", STD 56, RFC 2453, DOI 10.17487/RFC2453, November 1998, <https://www.rfc-editor.org/info/rfc2453>.

[RFC7868] Savage, D., Ng, J., Moore, S., Slice, D., Paluch, P., and R. White, "Cisco’s Enhanced Interior Gateway Routing Protocol (EIGRP)", RFC 7868, DOI 10.17487/RFC7868, May 2016, <https://www.rfc-editor.org/info/rfc7868>.

[RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <https://www.rfc-editor.org/info/rfc8200>.

11.2. Informative References

[I-D.lhan-problems-requirements-satellite-net]
Han, L., Li, R., Retana, A., Chen, M., Su, L., Jiang, T., and N. Wang, "Problems and Requirements of Satellite Constellation for Internet", Work in Progress, Internet-Draft, draft-lhan-problems-requirements-satellite-net-05, 5 July 2023, <https://datatracker.ietf.org/doc/html/draft-lhan-problems-requirements-satellite-net-05>.

[I-D.jia-flex-ip-address-structure]
Jia, Y., Chen, Z., and S. Jiang, "Flexible IP: An Adaptable IP Address Structure", Work in Progress, Internet-Draft, draft-jia-flex-ip-address-structure-00, 31 October 2020, <https://datatracker.ietf.org/doc/html/draft-jia-flex-ip-address-structure-00>.

[I-D.li-native-short-addresses]
Li, G., Jiang, S., and D. E. Eastlake, "Native Short Addresses for the Internet Edge", Work in Progress, Internet-Draft, draft-li-native-short-addresses-01, 25 May 2021, <https://datatracker.ietf.org/doc/html/draft-li-native-short-addresses-01>.

[I-D.lhan-satellite-instructive-routing]
Han, L., Retana, A., and R. Li, "Semantic Address Based Instructive Routing for Satellite Network", Work in Progress, Internet-Draft, draft-lhan-satellite-instructive-routing-02, 3 March 2023, <https://datatracker.ietf.org/doc/html/draft-lhan-satellite-instructive-routing-02>.

[I-D.retana-lsr-ospf-monitor-node]
Retana, A. and L. Han, "OSPF Monitor Node", Work in Progress, Internet-Draft, draft-retana-lsr-ospf-monitor-node-00, 7 March 2022, <https://datatracker.ietf.org/doc/html/draft-retana-lsr-ospf-monitor-node-00>.

[Routing-for-LEO]
Ekici, E., Akyildiz, I. F. and M. D. Bender, ""Datagram routing algorithm for LEO satellite networks," Proceedings IEEE INFOCOM 2000. Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No.00CH37064), 2000, pp. 500-508 vol.2, doi: 10.1109/INFCOM.2000.832223.", <https://ieeexplore.ieee.org/document/832223>.

[Length-Variable-IP]
Ren, S., Yu, D., Li, G., Hu, S., Tian, Y., Gong, X. and R. Moskowitz, ""Routing and Addressing with Length Variable IP Address," NEAT'19: Proceedings of the ACM SIGCOMM 2019 Workshop on Networking for Emerging Applications and Technologies, August 2019", <https://doi.org/10.1145/3341558.3342204>.

[IEEE-MAC-Address]
"IEEE Std 802-2001 (PDF). The Institute of Electrical and Electronics Engineers, Inc. (IEEE). 2002-02-07. p. 19. ISBN 978-0-7381-2941-9. Retrieved 2011-09-08.", <https://standards.ieee.org/getieee802/download/802-2001.pdf>.

[Lowest-LEO-ESA]
"Lowest LEO by ESA",
<https://www.esa.int/ESA_Multimedia/Images/2020/03/Low_Earth_orbit#:~:text=A%20low%20Earth%20orbit%20(LEO,very%20far%20above%20Earth's%20surface.>.

[KeplerianElement]
"Keplerian elements",
<https://en.wikipedia.org/wiki/Orbital_elements>.

[OSI-Model]
"OSI Model", <https://en.wikipedia.org/wiki/OSI_model>.

[StarLink] "Star Link", <https://en.wikipedia.org/wiki/Starlink>.

[China-constellation]
"China Constellation", <https://www.itu.int/ITU-R/space/asreceived/Publication/DisplayPublication/23706>.

[SpaceX-Non-GEO]
"FCC report: SPACEX V-BAND NON-GEOSTATIONARY SATELLITE SYSTEM", <https://fcc.report/IBFS/SAT-LOA-20170301-00027/1190019.pdf>.

[Large-Scale-LEO-Network-Routing]
Han, L.,Retana, A., Westphal, C. and R. Li, "Large Scale LEO Satellite Networks for the Future Internet: Challenges and Solutions to Addressing and Routing," Computer Networks and Communications, 1(1), 31-58", 2023, <https://ojs.wiserpub.com/index.php/CNC/article/view/2105>.

[Savi] "Satellite constellation visualization",
<https://savi.sourceforge.io/>.

Appendix A. Change Log

* Initial version, 10/19/2021

* 01 version, 02/28/2022

* 02 version, 09/02/2022

* 03 version, 03/03/2023

* 04 version, 09/01/2023

Authors' Addresses

Lin Han (editor)
Futurewei Technologies, Inc.
2330 Central Express Way
Santa Clara, CA 95050,
United States of America
Email: lhan@futurewei.com

Richard Li
Futurewei Technologies, Inc.
2330 Central Express Way
Santa Clara, CA 95050,
United States of America
Email: rli@futurewei.com

Alvaro Retana
Futurewei Technologies, Inc.
2330 Central Express Way
Santa Clara, CA 95050,
United States of America
Email: alvaro.retana@futurewei.com

Meiling Chen
China Mobile
32, Xuanwumen West
BeiJing 100053
China
Email: chenmeiling@chinamobile.com

Ning Wang
University of Surrey
Guildford
Surrey, GU2 7XH
United Kingdom
Email: n.wang@surrey.ac.uk

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: 5 October 2023

Z. Li
S. Peng
Huawei Technologies
D. Voyer
Bell Canada
C. Li
China Telecom
P. Liu
China Mobile
C. Cao
China Unicom
G. Mishra
Verizon Inc.
3 April 2023

Application-aware Networking (APN) Framework
draft-li-apn-framework-07

Abstract

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

This document proposes a new framework, named Application-aware Networking (APN), where application-aware information (i.e. APN attribute) including APN identification (ID) and/or APN parameters (e.g. network performance requirements) is encapsulated at network edge devices and carried in packets traversing an APN domain in order to facilitate service provisioning, perform fine-granularity traffic steering and network resource adjustment.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

A multitude of applications are carried over the network, which have varying needs for network bandwidth, latency, jitter, and packet loss, etc. Some applications such as online gaming and live video streaming have very demanding network requirements and therefore require special treatment in the network. However, in current networks, the network and applications are decoupled, that is, the network is not aware of the applications' requirements in a fine granularity. Therefore, it is difficult to provide truly fine-granularity traffic operations for the applications and guarantee their SLA requirements accordingly. [I-D.li-apn-problem-statement-usecases] describes the challenges of traditional differentiated service provisioning methods, such as five tuples used for ACL/PBR causing coarse granularity as well as orchestration and SDN-based solution causing long control loops.

This document proposes a new framework, named Application-aware Networking (APN), where application-aware information (APN attribute) including application-aware identification (APN ID) and application-aware parameters (APN Parameters), is encapsulated at network edge devices and carried along with the encapsulation of the tunnel used by the packet when traversing the APN domain. By APN domain we intend the operator infrastructure where APN is used from edge to edge (ingress to egress) and where the packet is encapsulated using an outer header incorporating the APN information. The APN attribute will facilitate service provisioning and provide fine-granularity services in the APN domain.

The APN attribute is acquired at the ingress of the APN domain based on the existing information in the incoming packet header (i.e. source and destination addresses, incoming L2 (or) MPLS encapsulation, incoming physical/virtual port information, the other fields of the 5-tuple if they are not encrypted) in the edge devices. The APN information is then added to the data packets along with the tunnel encapsulation. The packet traverses the domain and, when exiting the domain, the outer header along with the APN information is removed.

APN aims to leverage the ability to apply policies to traffic flows entering into the infrastructure (APN domain). For example, at the headend (ingress) traffic is steered into a given path/policy, at a midpoint node the corresponding performance measurement data is collected, and at a service node a given function is executed.

APN assumes traffic is tunnel encapsulated edge-to-edge tunnel encapsulation within a limited trusted domain. It means that the source and destination addresses of the packet are the endpoints of the tunnel (i.e. the domain edges), and nothing about the payload source and destination can be deduced, which substantially reduces the privacy concerns. Typically, an APN domain is defined as a Network Operator controlled limited domain (see Figure 1), in which MPLS, VXLAN, SR/SRv6 and other tunnel technologies are adopted to provide network services.

2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 RFC 2119 [RFC2119] RFC 8174 [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Terminology

ACL: Access Control List

APN: Application-aware Networking

APN6: Application-aware Networking for IPv6/SRv6

LB: Load Balancing

MPLS: Multiprotocol Label Switching

PBR: Policy Based Routing

QoE: Quality of Experience

SDN: Software Defined Networking

SLA: Service Level Agreement

SR: Segment Routing

SR-MPLS: Segment Routing over MPLS dataplane

SRv6: Segment Routing over IPv6 dataplane

4. APN Framework and Key Components

The APN framework is shown in Figure 1. The key components include App-aware Edge Device (APN-Edge), App-aware-process Head-End (APN-Head), App-aware-process Mid-Point (APN-Midpoint), App-aware-process End-Point (APN-Endpoint), and APN-Controller. The key interfaces include both the Northbound Interface (NBI) and the Southbound Interface (SBI) of the APN-Controller.

Packets carry application characteristic information (i.e. APN attribute) which includes the following information:

* Application-aware identification (APN ID): identifies the set of attributes, indicating that all packets belonging to the same flow will be given the same treatment by the network. APN ID is mandatory.

* Application-aware parameters (APN parameters): The typical application-aware parameters are the network performance requirement parameters including bandwidth, delay, delay variation, packet loss ratio, etc. APN parameters are optional.

```
                            +------------------+
                            |   APN-Customer   |
                            +------------------+
                                     |
                                     | NBI
                                     |
                            +------------------+
                     -------|  APN-Controller  |--------
                   /        +------------------+        \
Client            /        /          |           \        \           Server
+-----+          /        /           | SBI        \        \          +-----+
|App x|-\       |        |            |             |        |      /->|App x|
+-----+ |   +-----+ +-----+   +---------+   +--------+ +-----+  |  +-----+
         \->|APN  | |APN  |-A-|APN      |-A-|APN     | |APN  |->/
User side   |-    |-|-    |-B-|-        |-B-|-       |-|-    |
         /->|Edge | |Head |-C-|Midpoint |-C-|Endpoint| |Edge |->\
+-----+ |   +-----+ +-----+   +---------+   +--------+ +-----+  |   +-----+
|App y|-/      |------------------APN Domain--------------|      \->|App y|
+-----+                                                            +-----+
```

Figure 1: Framework and Key Components

The key components are introduced as follows.

* App-aware Edge Device (APN-Edge): this network device receives packets from applications and obtains the APN attribute based on the configuration on this device according to the existing information in the packet header, such as 5-tuple, VLAN or double VLAN tagging (C-VLAN and S-VLAN). The APN-Edge device adds the APN attribute in the tunnel encapsulation. The packets carrying the APN attribute will be sent to the APN-Head, and the APN attribute will be used to apply various policies in different nodes along the network path onto the traffic flow, e.g., at the headend to steer into corresponding path satisfying SLAs, at the midpoint to collect corresponding performance measurement data, at the service function to execute particular policies. When the packets leave the APN domain, the APN attribute will be removed together with the tunnel encapsulation.

* App-aware-process Head-End (APN-Head): This network device receives packets from the APN-Edge, obtains the APN attribute, and initiates the corresponding process. Generally, in order to satisfy different SLA requirements, a set of paths, tunnels or SR policies, are set up between the APN-Head and the APN-Endpoint. These multiple parallel paths have different SLA guarantees. The APN-Head maintains the matching relationship between the APN attribute and the paths between the APN-Head and the APN-Endpoint. The APN-Head determines the path between the APN-Head and the APN-Endpoint according to the APN attribute carried in the packets and the matching relationship with it, which satisfies the service requirements of the applications. The APN-Head forwards the packets along the path. The APN attribute conveyed by the packet received from the APN-Edge can also be copied or be mapped to the outgoing packet header.

* App-aware-process Mid-Point (APN-Midpoint): the APN-Midpoint provides the path service and enforces various policies according to the APN attribute carried in the packets. The APN-Midpoint may also adjust the resource locally to guarantee the service requirements depending on a specific policy and the APN attribute conveyed by the packet. Policy definitions and mechanisms are out of the scope of this document.

* App-aware-process End-Point (APN-Endpoint): the process of the specific service path will end at the APN-Endpoint. If the outer tunnel header for the path between the APN-Head and the APN-Endpoint exists, it will be removed by the APN-Endpoint. If the APN attribute is copied or mapped to the outer tunnel header by the APN-Head, it will also be removed along with the outer tunnel header.

Note that, the APN-Edge can co-exist with the APN-Head or APN-Endpoint, that is, one network device can implement the functionalities of both APN-Edge and the APN-Head/APN-Endpoint.

* APN-Controller: the APN-Controller is a functional block responsible for planning and execution of how the APN attribute are allocated and maintained. It collects the service requirements using a NBI (northbound interface) and signals the APN related policies to the network devices using SBI (southbound interfaces). To be more specific, the APN related policies include the APN-marking policy (i.e., the matching state in order for the node to classify and encapsulate the incoming packet with the desired APN Information) being pushed on the APN-Edge, and the APN-servicing policy (i.e., the APN state in order to fulfill the service requirements using e.g. traffic steering and performance measurement) on the APN-Head, APN-Midpoint, APN-Endpoint.

* APN-Customer: the APN-Customer manages the APN network services [I-D.li-apn-problem-statement-usecases] using the APN customer service model, which describes the requirements on the APN network services from the point of view of the APN customer.

The key interfaces are introduced as follows.

* The Northbound Interface (NBI) of the APN-Controller: the requirements described in the APN service model by the APN-Customer is enforced via this interface to the APN-Controller, which will be translated into the APN policies, i.e., the APN-marking policy and the APN-servicing policy, in the APN-Controller.

* The Southbound Interface (SBI) of the APN-Controller: the APN-marking policy and the APN-servicing policy are enforced via this interface from the APN-Controller to the relevant network devices. The candidate protocols for this interface are PCEP, BGP, YANG-based protocols (NETCONF/RESTCONF), etc.

These interfaces also include the operational status and monitoring information.

5. APN Requirements

This section specifies the requirements for supporting the APN framework, including the requirements for conveying and handling the APN attribute.

5.1. APN Attribute Conveying Requirements

The APN attribute consists of APN ID and APN parameters.

APN ID includes the following identifiers (IDs),

* Application Group ID: identifies an application group of the traffic.

* User Group ID: identifies the user group of the traffic.

APN ID can be acquired through different ways. In the APN framework it MUST be acquired according to the existing available information in the packet header without inspection of the payload.

The different combinations of the IDs can be used to provide different granularity of the service provisioning and SLA guarantee for the traffic.

The APN parameters are the network performance requirement parameters. The network service requirement can include the following parameters:

* Bandwidth: the bandwidth requirement

* Latency: the latency requirement

* Packet loss ratio: the packet loss ratio requirement

* Jitter: the jitter requirement

The different combinations of the parameters are for further expressing the more detailed service requirements, conveyed together with the APN ID, which can be used to match to appropriate tunnels/SR Policies and queues that can satisfy these service requirements.

APN attribute MUST be encapsulated within tunnels in the network layer. The tunnels include but not limit to MPLS, VxLAN, SR-MPLS, and SRv6. It can be extended according to requirements in the future.

[REQ 1a]. APN ID SHOULD include Application Group ID to indicate the application group that the packet belongs to.

[REQ 1b]. APN ID SHOULD include User Group ID to indicate the user group that the packet belongs to.

[REQ 1c]. APN ID MUST include either Application Group ID or User Group ID.

[REQ 1d]. APN ID MUST be acquired from the existing available information of the packet header without interference into the payload.

[REQ 1e]. APN parameters is OPTIONAL.

[REQ 1f]. APN attribute MUST be carried by the outer tunnel encapsulation.

[REQ 1g]. All the nodes along the path SHOULD be able to process the APN attribute if needed.

[REQ 1h]. The APN attribute is generated by the APN-Edge though local policy.

[REQ 1i]. The APN attribute SHOULD be kept intact when directly copied at the APN-Head and carried in the tunnel encapsulation.

[REQ 1j]. The APN attribute MUST be removed along with the tunnel encapsulation by the APN-Edge when the packets leave the APN domain.

[REQ 1k]. The APN attribute MUST NOT be encrypted when the APN packet is itself encrypted (e.g., the APN tunnel across the APN domain uses IPsec).

5.1.1. Protocol Extensions Requirements

The APN attribute is conveyed with the tunnel encapsulation. There are two typical types of tunnels:

* MPLS-based tunnel: LDP tunnel, RSVP-TE tunnel, SR-MPLS tunnel or policy, etc.

* IPv6-based tunnel: IPv6-based VxLAN tunnel, IPv6-based UDP tunnel, IPv6-based GRE tunnel, SRv6 tunnel or policy, etc.

In order to support encapsulation of APN attribute, the MPLS data plane and IPv6 data plane need to be extended.

In order to support acquiring the APN attribute according to the existing available information in the packet header, YANG models should be defined to configure the mapping between the application/ user group ID and the existing information in the packet header and configure the corresponding APN attribute for the application/user group. It can also be implemented with protocol extensions such as BGP/BGP-LS and PCEP which can advertise the information from the central controller to the APN-Edge.

In addition, in the APN domain, the above-mentioned mapping and applying APN parameters may also be advertised from the APN-Edge/APN-Head to other devices or from the network devices to the central controller in the APN domain. IGP extensions or BGP-LS extensions should be introduced to achieve the purposes.

[REQ 1-1a] MPLS encapsulation SHOULD be extended to be able to carry the APN attribute for MPLS-based tunnels.

[REQ 1-1b] IPv6 encapsulation SHOULD be extended to be able to carry the APN attribute for IPv6-based tunnels.

[REQ 1-1c] YANG models SHOULD be defined to implement the mapping between the application/user group ID and the existing available information in the packet header and configure the corresponding APN parameters.

[REQ 1-1d] BGP extensions SHOULD be defined to advertise the mapping between the application/user group ID and the existing available information in the packet header and the corresponding APN parameters from the central controller to the APN-Edge in the APN domain.

[REQ 1-1e] PCEP extensions SHOULD be defined to advertise the mapping between the application/user group ID and the existing available information in the packet header and the corresponding APN parameters from the central controller to the APN-Edge in the APN domain.

[REQ 1-1f] IGP extensions SHOULD be defined to advertise the mapping between the application/user group ID and the existing available information in the packet header and the corresponding APN parameters from the APN-Edge to the network devices in the APN domain.

[REQ 1-1g] BGP-LS extensions SHOULD be defined to advertise the mapping between the application/user group ID and the existing available information in the packet header and the corresponding APN parameters from the network devices to the central controller in the APN domain.

5.2. APN attribute Handling Requirements

The APN Head and APN-Midpoint perform matching operation against the APN attribute, that is, to match IDs and/or service requirements to the corresponding network resources such as tunnels/SR policies and queues.

5.2.1. Fine granular SLA Guarantee

In order to achieve better Quality of Experience (QoE) of end users and engage customers, the network needs to be able to provide fine-granularity SLA guarantee [I-D.li-apn-problem-statement-usecases].

[REQ 2-1a]. With the APN attribute, the APN-Head SHOULD be able to steer the traffic to the tunnel/SR policy that satisfies the matching operation.

[REQ 2-1b]. With the APN attribute, the APN-Head SHOULD be able to trigger the setup of the tunnel/SR policy that satisfies the matching operation.

[REQ 2-1c]. With the APN attribute, the APN-Head and APN-Midpoint SHOULD be able to steer the traffic to the queue that satisfies the matching operation.

[REQ 2-1d]. With the APN attribute, the APN-Head and APN-Midpoint SHOULD be able to trigger the configuration of the queue that satisfies the matching operation.

[REQ 2-1e]. If the tunnels are used to satisfy the performance requirements, the APN-Head SHOULD be able to copy or map the APN attribute conveyed by the packet received from the APN-Edge to the outer tunnel header.

[REQ 2-1f]. If the tunnels are used to satisfy the performance requirements and the APN attribute are conveyed along with the outer tunnel, the APN-Endpoint MUST remove the APN attribute along with the outer tunnel.

5.2.2. Fine granular network slicing

Network Slicing provides the ability to define a number of isolated network slices having different set of requirements.

APN is to help the operator of a network to steer some of the traffic tagged with an APN attribute to a certain network slice based on the SLA agreement with its customer.

[REQ 2-2a]. With the APN attribute, the APN-Head SHOULD be able to steer the traffic to the slice that satisfies the matching operation.

[REQ 2-2b]. With the APN attribute, the APN-Midpoint SHOULD be able to associate the traffic to the resources in the slice that satisfies the matching operation.

5.2.3. Fine granular deterministic networking

Along the path each node needs to provide guaranteed bandwidth, bounded latency, and other properties relevant to the transport of time-sensitive data for the Detnet flows that coexist with the best-effort traffic.

APN is to help the operator of a network to steer some of the traffic tagged with an APN attribute to a certain deterministic path based on the SLA agreement with its customer.

[REQ 2-3a]. With the APN attribute, the APN-Head MUST be able to steer the traffic to the appropriate path that satisfies the matching operation.

[REQ 2-3b]. With the APN attribute, the APN-Head MUST be able to trigger the setup of the appropriate path that satisfies the matching operation for the Detnet flows.

[REQ 2-3c]. With the APN attribute, the APN-Midpoint MUST be able to associate the traffic to the resources along the path that satisfies the performance guarantee.

[REQ 2-3d]. With the APN attribute, the APN-Midpoint MUST be able to reserve the resources for the Detnet flows along the path that satisfies the performance guarantee.

5.2.4. Fine granular service function chaining

The end-to-end service delivery often needs to go through various service functions, including traditional network service functions such as firewalls, LB as well as new application-specific functions, both physical and virtual. SFC is applicable to both fixed and mobile networks as well as data center networks.

APN is to help the operator of a network to steer some of the traffic tagged with an APN attribute to a certain service function chain based on the SLA agreement with its customer. On each service function along the service function chain, the policy can be enforced based on the APN attribute in the outer header.

[REQ 2-4a]. With the APN attribute, the App-aware-process devices SHOULD be able to steer the traffic to the appropriate service function.

[REQ 2-4b]. The App-aware-process devices including VAS SHOULD be able to process the APN attribute carried in the packets.

5.2.5. Fine granular network measurement

Network measurement can be used for verifying whether the network performance requirements have been satisfied, as well as locating silent failure and predicting QoE satisfaction, which enables real-time SLA awareness/proactive OAM and potential resource adjustments.

APN is to help the operator of a network to trigger performance measurement for the traffic tagged with an APN attribute based on its customer' consent.

[REQ 2-5a]. The App-aware-process devices SHOULD be able to perform IOAM based on the APN attribute.

[REQ 2-5b]. The network measurement results can be reported based on the APN attribute and verify whether the performance requirements are satisfied.

6. Illustration

In order to better clarify what APN can enable with the introduced APN attribute compared to the existing network without APN, we illustrate how APN works through an example use case, which is also a typical network service being provisioned nowadays, i.e. the Cloud Leased Line service. In order to make the tunnel description much easier to understand, we use the recent technology in IETF, i.e. SRv6.

6.1. Example use case description

We take the "SRv6-based Cloud Leased Line Service" as an illustrative example to show how APN is needed and can be beneficial.

Enterprises usually buy Cloud Leased Line Service to interconnect their local sites to Cloud. Generally, the Cloud Leased Line Service needs to go across multiple domains which are owned by the same operator and can be controlled by multiple controllers and an orchestrator/super-controller.

Due to management reasons, the network information in the intermediate domain cannot be advertised to other domains, so the ingress node cannot set up an appropriate E2E path. In that case, the intermediate domain is treated as a black box, and no fine grain traffic steering and other services can be provisioned.

The example of the network to provide the cloud leased lined service reference diagram is shown as the following figure. The network is composed by three network domains including the two metro networks in the City A and City B and the backbone network which connects the two metro networks. The cloud leased line services is provided to the specific enterprise whose branches located in different cities need to access the cloud-based service located in the City B.

```
              Domain 1                 Domain 2              Domain 3
           /-------------\         /------------\         /-----------\
         +-/-+   City A  +-\-+   +-/-+   IPv6   +-\-+   +-/-+ City B  +-\-+
Branch---|PE1|           |CR1|---|CR2|          |CR3|---|CR4|         |PE2|--Clou
d
         +-\-+    Metro  +-/-+   +-\-+ Backbone +-/-+   +-\-+  Metro  +-/-+
           \-------------/         \------------/         \-----------/

          |<--OSPF/ISIS-->|<-EBGP->|<- IPv6/SRv6 ->|<-EBGP->|<-OSPF/ISIS->|
          |<------------------------ EBGP VPNv4 Peer ---------------------->|
          |<------------------------ L3VPN over SRv6 ---------------------->|
```

Figure 2: Reference diagram for the example use case illustration

6.2. User Group and Application Group Design

The user groups can be designed as follows:

	User Group
Enterprise A/Branch 1/Office Users	001001001
Enterprise A/Branch 1/R&D Users	001001002
Enterprise A/Branch 1/IT Users	001001003
Enterprise A/Branch 1/VIP Users	001001004
Enterprise A/Branch 2/Office Users	001002001
Enterprise A/Branch 2/R&D Users	001002002
Enterprise A/Branch 2/IT Users	001002003
Enterprise A/Branch 2/VIP Users	001002004
Enterprise A/Branch 3/Office Users	001003001
Enterprise A/Branch 3/R&D Users	001003002
Enterprise A/Branch 3/IT Users	001003003
Enterprise A/Branch 3/VIP Users	001003004

In the IP address design, the IPv6 address blocks allocated to the branches are as follows :

	IPv6 Address
Enterprise A/Branch 1/Office Users	2001:DB8:A:11::/56
Enterprise A/Branch 1/R&D Users	2001:DB8:A:12::/56
Enterprise A/Branch 1/IT Users	2001:DB8:A:13::/56
Enterprise A/Branch 1/VIP Users	2001:DB8:A:1D::/56
Enterprise A/Branch 2/Office Users	2001:DB8:A:21::/56
Enterprise A/Branch 2/R&D Users	2001:DB8:A:22::/56
Enterprise A/Branch 2/IT Users	2001:DB8:A:23::/56
Enterprise A/Branch 2/VIP Users	2001:DB8:A:2D::/56
Enterprise A/Branch 3/Office Users	2001:DB8:A:31::/ 56
Enterprise A/Branch 3/R&D Users	2001:DB8:A:32::/56
Enterprise A/Branch 3/IT Users	2001:DB8:A:33::/56
Enterprise A/Branch 3/VIP Users	2001:DB8:A:3D::/56

The application groups provided by the cloud can be designed as follows:

	Application Group
Enterprise A/Office Audio Applications	101001001
Enterprise A/Office Video Applications	101001002
Enterprise A/Office Data Applications	101001003
Enterprise A/R&D Audio Applications	101002001
Enterprise A/R&D Video Applications	101002002
Enterprise A/R&D Data Applications	101002003
Enterprise A/IT Audio Applications	101003001
Enterprise A/IT Video Applications	101003002
Enterprise A/IT Data Applications	101003003

In the address design, the IPv6 address blocks allocated to the applications of Enterprise A in the cloud is 2001:DB8:A1::/48A1::A:/16. The port number can be used to identify different applications.

	IPv6 Address	Port Number
Enterprise A/Office Audio Applications	2001:DB8:A1:A1::/64	1718, 1719
Enterprise A/Office Video Applications	2001:DB8:A1:A1::/64	5060, 5061
Enterprise A/Office Data Applications	2001:DB8:A1:A1::/64	21, 80
Enterprise A/R&D Audio Applications	2001:DB8:A1:A2::/64	1718, 1719
Enterprise A/R&D Video Applications	2001:DB8:A1:A2::/64	5060, 5061
Enterprise A/R&D Data Applications	2001:DB8:A1:A2::/64	21, 80
Enterprise A/IT Audio Applications	2001:DB8:A1:A3::/64	1718, 1719
Enterprise A/IT Video Applications	2001:DB8:A1:A3::/64	5060, 5061
Enterprise A/IT Data Applications	2001:DB8:A1:A3::/64	21, 80

6.3. Derive the User Group and User Group at APN Edge

The cloud leased line service adopts the SRv6-based L3VPN to traverse the network. The following policy can be applied at the APN edges of the City A1:

```
Match:
  VPN1
  Source Address 2001:DB8:A:11::/56
Action
  Set user-group 001001001

Match:
  VPN1
  destination Address 2001:DB8:A1:A1::/64
  destination port 1718,1719
Action
  Set app-group 101001001
```

6.4. Access Right Check at the edge of the backbone network

The following check can be applied at the edge of the IP backbone network:

```
Match:
  user-group 001001001
  app-group  101002001, 101002002, 101002003, 101003001, 101003002, 101003003
Action
  Deny

Match:
  user-group 001001001
  app-group  101001001, 101001002, 101001003
Action
  Permit
```

The policy means that the office users of the branch 1 can only access the office applications.

If the address allocation is changed. For example, one office user of the branch1s IPv6 address is changed to 2001:DB8:A:15::/56 because of the mobile office.

We only need to add the following policy at the APN edge:

```
Match:
  VPN1
  Source Address 2001:DB8:A:15::/56
Action
  Set user-group 001001001
```

The policy in the backbone network which is based on the user group and the application group is not necessary to change.

6.5. SLA Guarantee in the backbone network

Due to management reasons, the network information in the intermediate domain cannot be advertised to other domains, so the ingress node cannot set up an appropriate TE path, the intermediate domain is treated as a black box and no fine grain traffic steering can be performed.

In this case, we consider fine grain traffic steering in Domain 2 on top of the SRv6-based Cloud Leased Line Service for the purpose of SLA Guarantee.

6.5.1. Network Measurement

In order to guarantee SLA for the VIP users, the following network measurement policy can be applied in the backbone network:

```
Match:
  User-group 001001004 application group 101001002
  User-group 001002004 application group 101002002
  User-group 001003004 application group 101003002
Action
  Apply IOAM
```

The policy is to apply the IOAM as the network measurement for the VIP users of the branches to access the video applications. From the above illustration, there is the following observation:

When there is no APN deployed, at CR2, the 5 tuples of the original packets will need to be resolved since they have been encapsulated, and then IOAM can be triggered based on the 5 tuples. This resolution process is costly and consumes a lot of hardware resources. If Domain 3 needs to trigger IOAM, the same resolution process will have to be done at CR4.

When there is APN deployed, at CR1, the APN attribute is tagged. When these packets arrive at CR2, only the APN attribute in the outer header will be read out, based on which the IOAM can be triggered in Domain 2. That is, no 5-tuple resolution process is needed at CR2 but only checking the APN attribute in the outer header.

6.5.2. Traffic Steering

If the SLA guarantee of the VIP users accessing the video applications does not satisfy the requirements through the network measurement based on the IOAM, the SRv6 policy can be setup. For example, the SRv6 policy 1 which can satisfy the SLA requirement is set up. Then the following policy can be downloaded to the edge:

```
Match:
  User-group 001001004 application group 101001002
  User-group 001002004 application group 101002002
  User-group 001003004 application group 101003002
Action
  Redirect SRv6 Policy 1
```

The policy is to steer the traffic of the VIP users to the SRv6 policy in the backbone to satisfy the requirement .

From the above illustration, there is the similar observation as the network measurement:

When there is no APN deployed, at CR2, the 5 tuples of the original packets will need to be resolved since they have been encapsulated, and then the traffic can be steered into SRv6 policy 2 based on the 5 tuples. This resolution process is costly and consumes a lot of hardware resources.

When there is APN deployed, at CR1, the APN attribute is tagged When these packets arrive at CR2, only the APN attribute in the outer header will be read out, based on which the traffic can be steered into SRv6 policy 2 in Domain 2.

7. Benefits

The APN attribute allows the network devices to only look at one easily-accessible field in the outer header, without having to resolve the 5 tuples of the original packets that are deeply encapsulated in the tunnel encapsulation.

The APN attribute allows to simplify the policy control at every policy enforcement point within the network. The APN attribute allows to reducing each matching entry of policy filter since it is

only one field and hardware resources are saved. Since APN attribute is relatively stable, it introduces the possibilities of eliminating the "stale" policy filter entries. In most cases, the APN attribute is centralized configured and distributed to all the policy enforcement points, which saves the policy filter configurations per node and simplifies the OM.

The structured APN attribute allows to express fine granular service requirements, e.g. MKT-user-group/app-group, RD-user-group/app-group, latency.

The structured APN attribute allows to match to the evolving fine granular differentiated network capabilities, e.g. SR policy with low latency and high reliability guaranteed.

In a tunnel across multiple domains of the same operator using the APN attribute in the outer header the operator can easily support multiple services not just a single one in a particular domain as illustrated in the use case illustration section.

When there is no APN, to achieve the same, now the operator may have two options: 1. Add all the policy identifiers at the tunnel headend with various further encapsulations and enforce the policies based on them at the intermediate policy enforcement nodes along the tunnel, 2. Resolve the original 5 tuples being encapsulated inside the tunnel which will be very costly and sometimes impossible.

Moreover, the policy enforcement table in the intermediate policy enforcement nodes is significantly reduced. Because before operator needs to resolve the 5 tuple but now with APN, operator only needs to read the APN attribute in one field of the outer header.

Since the 5 tuples of the traffic are changing frequently due to service deployment or management issues the policy enforcement table in the policy enforcement nodes is not stable and there is always a lot of stale entries in the table. But now since the APN attribute is a mapping of the 5 tuples operator will have a relatively stable policy enforcement table on their nodes.

8. IANA Considerations

This document does not include an IANA request.

9. Security Considerations

In the APN work, in order to reduce the privacy and security issues, the following specifications are defined:

[S1]. The APN attribute MUST be conveyed along with the tunnel information in the APN domain. The APN attribute is encapsulated and removed at the APN-Edge.

[S2]. The APN ID (including the Application Group ID and the User Group ID) MUST be acquired from the existing available information in the packet header without interference into the payload.

According to the above specifications, the APN attribute is only produced and used locally within the APN domain without the involvement of the host/application side.

In order to prevent the malicious attack through the APN attribute, the following policies can be configured at the network devices of the APN domain:

[P1]. If the APN attribute is conveyed without the tunnel information, the packet MUST be dropped.

[P2]. If the APN attribute is not known to the APN domain, it should trigger the alarm information. The packet can be forwarded without being processed or dropped depending on the local policy.

[P3]. If the network service requirements exceed the specification for the specific Application Group ID and/or User Group ID, it should trigger the alarm information. The packet should be discarded to prevent abusing of the resources.

[P4]. There should be rate-limiting policy at the APN-Edge to prevent the traffic belonging to a specific Application Group ID and/or User Group ID from exceeding the preset limit.

10. Acknowledgements

The authors would like to acknowledge Robert Raszuk (Bloomberg LP), Yukito Ueno (NTT Communications Corporation), and Dhruv Dhody for their valuable reviews and comments.

11. Co-authors

Kentaro Ebisawa
Toyota Motor Corporation
Japan

Email: ebisawa@toyota-tokyo.tech

Stefano Previdi
Huawei Technologies
Italy

Email: stefano@previdi.net

James N Guichard
Futurewei Technologies Ltd.
USA

Email: jguichar@futurewei.com

12. Contributors

Daniel Bernier
Bell Canada

Email: daniel.bernier@bell.ca

Chongfeng Xie
China Telecom

Email: xiechf@chinatelecom.cn

Feng Yang
China Mobile

Email: yangfeng@chinamobile.com

Zhuangzhuang Qin
China Unicom

Email: qinzhuangzhuang@chinaunicom.cn

Chang Liu
China Unicom

Email: liuc131@chinaunicom.cn

Gyan Mishra
Verizon

Email: hayabusagsm@gmail.com

Luis M. Contreras
Telefonica

Email: contreras.ietf@gmail.com

Luc-Fabrice Ndifor Ngwa
MTN

Email: Luc-Fabrice.Ndifor@mtn.com

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>.

[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>.

13.2. Informative References

[I-D.li-apn-problem-statement-usecases] Li, Z., Peng, S., Voyer, D., Xie, C., Liu, P., Qin, Z., and G. Mishra, "Problem Statement and Use Cases of Application-aware Networking (APN)", Work in Progress, Internet-Draft, draft-li-apn-problem-statement-usecases-08, 3 April 2023, <https://datatracker.ietf.org/api/v1/doc/document/draft-li-apn-problem-statement-usecases/>.

Authors' Addresses

Zhenbin Li
Huawei Technologies
China
Email: lizhenbin@huawei.com

Shuping Peng
Huawei Technologies
China
Email: pengshuping@huawei.com

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

Cong Li
China Telecom
China
Email: licong@chinatelecom.cn

Peng Liu
China Mobile
China
Email: liupengyjy@chinamobile.com

Chang Cao
China Unicom
China
Email: caoc15@chinaunicom.cn

Gyan Mishra
Verizon Inc.
United States of America
Email: gyan.s.mishra@verizon.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: 5 October 2023

Z. Li
S. Peng
Huawei Technologies
D. Voyer
Bell Canada
C. Xie
China Telecom
P. Liu
China Mobile
Z. Qin
China Unicom
G. Mishra
Verizon Inc.
3 April 2023

Problem Statement and Use Cases of Application-aware Networking (APN)
draft-li-apn-problem-statement-usecases-08

Abstract

Network operators are facing the challenge of providing better network services for users. As the ever-developing 5G and industrial verticals evolve, more and more services that have diverse network requirements such as ultra-low latency and high reliability are emerging, and therefore differentiated service treatment is desired by users. On the other hand, as network technologies such as Hierarchical QoS (H-QoS), SR Policy, and Network Slicing keep evolving, the network has the capability to provide more fine-granularity differentiated services. However, network operators are typically unaware of the applications that are traversing their network infrastructure, which means that not very effective differentiated service treatment can be provided to the traffic flows. As network technologies evolve including deployments of IPv6, SRv6, Segment Routing over MPLS dataplane, the programmability provided by IPv6 and Segment Routing can be augmented by conveying application related information into the network satisfying the fine-granularity requirements.

This document analyzes the existing problems caused by lack of service awareness, and outlines various use cases that could benefit from an Application-aware Networking (APN) framework.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

Due to the requirement for differentiated traffic treatment driven by diverse new services, the ability to convey the application-aware information and program the network infrastructure accordingly to provide fine-grained services is becoming increasingly necessary for network operators. The Application-aware Networking (APN) framework is being defined to address the requirements and some use cases are described in this document. APN takes advantage of network programmability by conveying application related information in the data plane so to facilitate network operators in providing fine-grained services.

One of the key objectives of APN is for network operators to provide fine-granularity SLA guarantees instead of coarse-grain traffic operations. This will allow to provide differentiated services for different applications and increase revenue accordingly. Among various applications being carried and running in the network, some revenue-producing applications such as online gaming, video streaming, and enterprise video conferencing have much more demanding performance requirements such as low network latency and high bandwidth. In order to achieve better Quality of Experience (QoE) for end users and engage customers, the network needs to be able to provide fine-granularity and even application group-level SLA guarantee.

2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 RFC 2119 [RFC2119] RFC 8174 [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Terminology

ACL: Access Control List

APN: Application-aware Networking

APN6: Application-aware Networking for IPv6/SRv6

DPI: Deep Packet Inspection

PBR: Policy Based Routing

QoE: Quality of Experience

SDN: Software Defined Networking

SLA: Service Level Agreement

MPLS: Multiprotocol Label Switching

SR: Segment Routing

SRv6: Segment Routing over IPv6 dataplane

SR-MPLS: Segment Routing over MPLS dataplane

VPN: Virtual Private Network

TE: Traffic Engineering

FRR: Fast Reroute

CAPEX: Capital expenditures

OPEX: Operating expenditures

4. Problem Statement

This section summarizes the challenges currently faced by network operators when attempting to provide fine-grained traffic operations to satisfy the various requirements demanded by new applications that require differentiated service treatment.

4.1. Challenges of lack of fine-granularity service information

In today’s networks, the infrastructure through which the traffic is forwarded is not able to provide fine-granularity SLAs satisfying specific requirements. It is therefore difficult for network operators to provide fine-grained traffic operations for various performance-demanding applications. In order to satisfy the SLA requirements network operators continue to increase the network bandwidth but only carrying very light traffic load (in general, around 30%-40% of its capacity).

As network technologies keep evolving, the network capability has been greatly enhanced and is able to provide fine-granularity service provisioning. For example,

H-QoS: provides hierarchical fine-grained QoS services.

SR Policy: provides the ability to handle a large number of explicit and flexible SR paths in order for services to select the policy that satisfies their SLA requirements.

Network Slicing: provides the ability to define a number of isolated network slices having different set of requirements.

IOAM: provides more accurate performance measurement of the traffic flow.

In summary, driven by the ever-emerging diverse demanding services, the lack of the fine-granularity information about the services in the network will cause the following issues: 1) the service information is not clearly described and known by the network, 2) the fine-granularity service provisioning capability is not fully utilized, 3) a fine-granularity service scheduling and measurement cannot be achieved.

4.2. Challenges of Traditional Differentiated Service Provisioning

The traditional ways used to provide fine-grained service provisioning face some challenges. The network devices mainly rely on the 5-tuple of the packets or DPI. However, there are some challenges for these traditional methods in differentiated service provisioning:

1. Five Tuples used for ACL/PBR: five tuples are widely used for ACL/PBR matching of traffic. However, these features cannot provide enough information for the fine-grained service process, and can only provide indirect application-level information which needs to be translated. Generally, ACLs involve high overhead on the forwarding process. Moreover, in some cases such as tunnel encapsulation and IPv6 data plane with a list of extension headers, it becomes impossible to resolve the 5 tuples due to the transport layer information being pushed very deep in the packet.

2. Deep Packet Inspection (DPI): If more information is needed, it must be extracted using DPI which can inspect deep into the packets for application specific information. However, this will introduce more CAPEX and OPEX for the network operator and impose security and privacy challenges.

3. Orchestration and SDN-based Solution: In the era of SDN, typically, an SDN controller is used to manage and operate the network infrastructure and orchestrator elements introduce application requirements so that the network is programmed accordingly. The SDN controller can be aware of the service requirements of the applications on the network through the interface with the orchestrator, and the service requirement is used by the controller for traffic management over the network. However, this method raises the following problems:

   a. The whole loop is long and time-consuming which is not suitable for fast service provisioning for critical applications;

   b. Too many interfaces are involved in the loop, as shown in Figure 1, which introduce challenges of standardization and inter-operability.

```
                  +--------------+
             +-----| Orchestrator | -------------------+
             |     +--------------+   Resource         |
   APP Req.  |             |         Management        |
           +---------+  +---------+       &        +---------+
           |SDN Ctrl1|  |SDN Ctrl2|    Service     |SDN Ctrl3|
           +---------+  +---------+  Provisioning  +---------+
   App Req./    |         |       \                   |       \
          /     |         |        \                  |        \
         /      |         |         \                 |         \
    +---+   +-----+   +--------+  +-------+   +-------+  +-------+
    |APP|   | DCN |   |Network |..|Network|   |Network|..|Network|
    +---+   +-----+   |   D1   |  |   D3  |   |   D4  |  |   D6  |
                      +--------+  +-------+   +-------+  +-------+
```

Figure 1: Multiple interfaces involved in the long service-provisioning loop

In [I-D.peng-apn-scope-gap-analysis], some mechanisms that have been specified in IETF and using attribute/identifier to perform traffic steering and service provisioning are analyzed. The existing solutions are specific to a particular scenario or data plane, and a generalized method used for fine-grained service provisioning is still missing.

4.3. Challenges of Supporting New 5G and Edge Computing Technologies

New technologies such as 5G, IoT, and edge computing, are continuously developing leading to more and more new types of services accessing the network. Large volumes of network traffic with diverse requirements such as low latency and high reliability are therefore rapidly increasing. If traditional methods for differentiation of traffic continue to be utilized, it will cause much higher CAPEX and OPEX to satisfy the ever-developing applications' diverse requirements.

5. Key Elements of Application-aware Networking (APN)

Application-aware Networking (APN) aims to address the problems mentioned in Section 4, associated with fine-grained traffic operations that are required in order to satisfy the various application-awareness requirements demanded by new services that need differentiated service treatment.

In APN, the application-aware information (APN Attribute) is derived according to the existing information in the packet header and encapsulated along with the encapsulation of the tunnel. With the APN attribute, fine-granularity network services can be provisioned within the APN domain accordingly. The APN attribute can include application-aware ID (APN ID) and application-aware parameters (APN Parameters). The APN ID can be derived, and applied to the packet, through a mapping mechanism leveraging the existing information in the packet header. The APN parameters can be applied for the corresponding APN ID according to the local policy. The typical APN parameters are the network performance requirements.

APN has the following key elements:

1. Application-aware information (APN attribute) is conveyed in the data plane through augmentation of existing encapsulations such as IPv6, SRv6 and MPLS. The conveyed APN attribute includes APN ID and/or APN parameters. This information is acquired at the edge of the APN domain according to the existing information in the packet header. When a data packet uses APN and conveys the application-aware information, it is referred in this document as an APN packet.

2. Application-aware information and network service provisioning matching providing fine-granularity network service provisioning (traffic operations) and SLA guarantee based on the APN attribute carried in APN packets. According to the APN attribute, appropriate network services are selected, provisioned, and provided to the demanding applications to satisfy their service requirements.

3. Measurement of the network performance so to maintain the match between the applications requirements and corresponding network services for a better fine-granularity SLA compliance. The network measurement methods include in-band and out-of-band, passive, active, per-packet, per-flow, per node, end-to-end, etc. These methods can also be integrated.

```
                          APN Network

Element 1: Conveying  ------------------->
                                /|\
            APN attribute        |     Network capabilities
                                 |       (SLA guarantee)
                                 |             /|\
                      Element 2: Matching      |
                                               |
                                     Element 3: Network Measurement
```

Figure 2: Illustration of the key elements of APN

6. Scenarios of APN Domains

1. SD-WAN scenario

The SD-WAN scenario is shown in the following figure. With APN, at the edge node, i.e. CPE, of the SD-WAN, the 5-tuple, plus information related to user or application group-level requirements is constructed into the APN attribute. When the packet is sent from the CPE, the attribute is added along with the tunnel encapsulation. This attribute is only meaningful for the network operators to apply various policies in different nodes/service functions, which can be enforced from the Controllers.

```
                          +-----------------+
                +---------|SD-WAN Controller|---------+
                |         +--------|--------+         |
                |                  |                  |
                |          +-------|-------+          |
                |          |SDN  Controller|          |
                |          +-------|-------+          |
   +-----+      |                  |                  |      +-----+
   |App x|-\    |                  |                  |    /-|App x|
   +-----+ |  +--|--+   +----------|----------+   +--|--+ | +-----+
            \-|     |   |  Application-aware   |   |     |-/
              |CPE 1|---|       Network        |---|CPE 2|
            /-|     |   |  Service Provisioning |   |     |-\
   +-----+ |  +-----+   +-----------------------+   +-----+ | +-----+
   |App y|-/    |                                  |    \-|App y|
   +-----+      |<---         APN Domain         --->|      +-----+
```

Figure 3: APN Domain in the Scenario of SD-WAN

2. Home broadband scenario

In the home broadband scenario, generally a home broadband user is authorized by the BNG. If the validation is passed and the access control is released, the user group can start enjoying the value-added service. With APN, when the traffic traverses the metro network, the traffic flow can be indicated by the APN attribute that is added/removed at the edge devices of the Metro Network (APN domain) based on the mapping from the existing information (e.g. the QinQ which is composed of C-VLAN and S-VLAN) in the packet header and then carried in the tunnel encapsulation header. The APN attribute will facilitate the fine-granular service in the APN domain. Once the packets leave the APN domain, the APN attribute will be removed together with the tunnel encapsulation header.

```
                                 |---- APN Domain ---|
  +----+                                .-----.
  | PC | \                             (       )
  +----+  \--\                      .--(         )--.
    +-----+   \+----+  +----+     (                 )      +-------+
    | STB |----| RG |--| AN |----(   Metro Network   )-----|  BNG  |--->
    +-----+   /+----+  +----+     (                 )      +-------+
  +-----+ /--/                     '--(         )--'
  |Phone|/                             (       )
  +-----+                              '-----'
                            QinQ                     QinQ
                           |----|----   Tunnel  ----|----|
```

Figure 4: Home Broadband Scenario

3. Mobile broadband scenario

In the mobile broadband scenario, a UE is authorized by the 5GC function, and the traffic steering and QoS policy are enforced by the UPF (User Plane Function) node. If the validation is passed and the access control is released, the user can start enjoying the value-added service. With APN, when the traffic traverses the mobile transport network, the traffic flow can be indicated by the APN attribute that is added at the edge devices of the mobile transport network (APN domain) based on mapping from the existing information (e.g. GTP-u tunnel encapsulation information) in the packet header and then carried in the tunnel encapsulation header. The APN attribute will facilitate the fine-granular service in the APN domain. Once the packets leave the APN domain, the APN attribute will be removed together with the tunnel encapsulation header.

```
                            |--  APN Domain  ---|

 +----+                            .-----.
 | PC |                          (       )
 +----+                       .--(         )--.
   +----+     +------+       (                 )       +-------+
   | UE | --- | gNB  |------(  Mobile Transport )------|  UPF  |---->
   +----+     +------+       (     Network     )       +-------+
 +-----+                      '--(         )--'
 | CPE |                          (       )
 +-----+                           '-----'

                             |----  Tunnel  ----|

                   |---------      GTP-u Tunnel     --------|
```

Figure 5: Mobile Broadband Scenario

7. Use cases for Application-aware Networking (APN)

This section illustrates some of the use cases that can benefit from APN. The corresponding requirements for APN are also outlined.

7.1. Application-aware SLA Guarantee

As stated in section 1, among various applications being carried and running in the network, some revenue-producing applications such as online gaming, video streaming, and enterprise video conferencing have much more demanding performance requirements such as low network latency and high bandwidth. In order to achieve better Quality of Experience (QoE) for end users and engage customers, the network needs to be able to provide fine-granularity and even application group-level SLA guarantee. Differentiated service provisioning is also desired.

The APN framework MUST address the following requirements:

* APN needs to perform the three key elements as described in Section 5.

* Support application group-level fine-granularity traffic operation that may include finer QoS scheduling.

7.2. Application-aware network slicing

More and more applications/services with diverse requirements are being carried over and sharing a common operators' network infrastructure. However, it is still desirable to have customized network transport that can support some applications' specific requirements, taking into consideration service and resource isolation, which drives the concept of network slicing.

Network slicing provides ways to partition the network infrastructure in either the control plane or data plane into multiple network slices that are running in parallel. These network slices can serve diverse services and fulfill their various requirements at the same time. For example, the mission critical application that requires ultra-low latency and high reliability can be provisioned over a separate network slice.

The APN framework MUST address the following requirements:

* APN needs to perform the three key elements as described in Section 5 in the context of network slicing.

* For the element 2, the APN framework MUST allow to assign a given traffic flow to specific network slice according to the APN attribute carried in the APN packet.

* For the element 3, the APN framework MUST allow the network measurement of each network slice.

7.3. Application-aware Deterministic Networking

[RFC8578] documents use cases for diverse industry applications that require deterministic flows over multi-hop paths. Deterministic flows provide guaranteed bandwidth, bounded latency, and other properties relevant to the transport of time-sensitive data, and can coexist on an IP network with best-effort traffic. It also provides for highly reliable flows through provision for redundant paths.

The APN framework MUST address the following requirements:

* APN needs to perform the three key elements as described in Section 5 in the context of deterministic networking.

* For the element 2, the APN framework MUST allow to assign a given traffic flow to a specific deterministic path according to the APN attribute carried in the APN packet.

* For the element 3, the APN framework MUST allow the network measurement of each application-aware deterministic path.

7.4. Application-aware Service Function Chaining

End-to-end service delivery often needs to go through various service functions including traditional network service functions such as firewalls, DPIs as well as new application-specific functions, both physical and virtual. The definition and instantiation of an ordered set of service functions and subsequent steering of the traffic through them is called Service Function Chaining (SFC) [RFC7665]. SFC is applicable to both fixed and mobile networks as well as data center networks.

Generally, in order to manipulate a specific traffic flow along the SFC, a DPI needs to be deployed as the first service function of the chain to detect the application, which will impose high CAPEX and consume long processing time. For encrypted traffic, it even becomes impossible to inspect the traffic flow.

The APN framework MUST address the following requirements:

* APN needs to perform the three key elements as described in Section 5 in the context of service function chaining.

* For the element 1, class information can be conveyed.

* For the element 2, the APN framework MUST allow to assign a given traffic flow to a specific service function chain and MUST allow the subsequent steering according to the APN attribute carried in the APN packets.

* For the element 3, the APN framework MUST allow the network measurement of each application-aware service function chain.

7.5. Application-aware Network Measurement

Network measurement can be used for locating silent failure and predicting QoE satisfaction, which enables real-time SLA awareness/ proactive OAM. Operations, Administration, and Maintenance (OAM) refers to a toolset for fault detection and isolation, and network performance measurement. In-situ Operations, Administration, and Maintenance (IOAM) records operational and telemetry information in the packet while the packet traverses a path between two points in the network.

The APN framework MUST address the following requirements:

* APN needs to perform the two key elements as described in Section 5 in the context of network measurement. The network measurement in the element 3 does not need to be considered here.

8. IANA Considerations

This document does not include an IANA request.

9. Security Considerations

In the APN work, in order to reduce the privacy and security issues, the APN attribute MUST be conveyed along with the tunnel information in the APN domain. The APN attribute is encapsulated and removed at the edge of the APN domain. The APN ID MUST be acquired from the existing available information in the packet header without interference into the payload.

According to the above specifications, the APN attribute is only produced and used locally within the APN domain without the involvement of the host/application side.

In order to prevent the malicious attack through the APN attribute, the following policies can be configured at the network devices of the APN domain. If the APN attribute is conveyed without the tunnel information, the packet MUST be dropped. If the APN attributes are not known to the APN domain, it should trigger the alarm information. The packet can be forwarded without being processed or dropped

depending on the local policy. If the network service requirements exceed the specification for the specific APN ID, it should trigger the alarm information. The packet should be discarded to prevent abusing of the resources. There should be rate-limiting policy at the edge of the APN domain to prevent the traffic belonging to a specific APN ID from exceeding the preset limit.

10. Acknowledgements

The authors would like to acknowledge Robert Raszuk (Bloomberg LP) and Yukito Ueno (NTT Communications Corporation) for their valuable review and comments.

11. Co-authors

Kentaro Ebisawa
Toyota Motor Corporation
Japan

Email: ebisawa@toyota-tokyo.tech

Stefano Previdi
Huawei Technologies
Italy

Email: stefano@previdi.net

James N Guichard
Futurewei Technologies Ltd.
USA

Email: jguichar@futurewei.com

12. Contributors

Daniel Bernier
Bell Canada
Canada

Email: daniel.bernier@bell.ca

Liang Geng
China Mobile
China

Email: gengliang@chinamobile.com

Chang Cao
China Unicom
China

Email: caoc15@chinaunicom.cn

Chang Liu
China Unicom
China

Email: liuc131@chinaunicom.cn

Cong Li
China Telecom
China

Email: licong@chinatelecom.cn

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>.

[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>.

[RFC8578] Grossman, E., Ed., "Deterministic Networking Use Cases", RFC 8578, DOI 10.17487/RFC8578, May 2019, <https://www.rfc-editor.org/info/rfc8578>.

[RFC7665] Halpern, J., Ed. and C. Pignataro, Ed., "Service Function Chaining (SFC) Architecture", RFC 7665, DOI 10.17487/RFC7665, October 2015, <https://www.rfc-editor.org/info/rfc7665>.

13.2. Informative References

[I-D.peng-apn-scope-gap-analysis] Peng, S., Li, Z., and G. S. Mishra, "APN Scope and Gap Analysis", Work in Progress, Internet-Draft, draft-peng-apn-scope-gap-analysis-06, 13 March 2023, <https://datatracker.ietf.org/doc/html/draft-peng-apn-scope-gap-analysis-06>.

Authors' Addresses

Zhenbin Li
Huawei Technologies
China
Email: lizhenbin@huawei.com

Shuping Peng
Huawei Technologies
China
Email: pengshuping@huawei.com

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

Chongfeng Xie
China Telecom
China
Email: xiechf@chinatelecom.cn

Peng Liu
China Mobile
China
Email: liupengyjy@chinamobile.com

Zhuangzhuang Qin
China Unicom
China
Email: qinzhuangzhuang@chinaunicom.cn

Gyan Mishra
Verizon Inc.
United States of America
Email: gyan.s.mishra@verizon.com

Network Working Group
Internet Draft
Intended status: Informational
Expires: January 6, 2023

Yisong Liu
W. Cheng
China Mobile
C. Lin
New H3C Technologies
X. Geng
Huawei Technologies
Y. Liu
ZTE

July 6, 2022

Considerations for Protection of SRv6 Networks
draft-liu-rtgwg-srv6-protection-considerations-02

Abstract

This document describes the considerations for protection of SRv6 network.

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction

Segment Routing [RFC8402] instantiated on the IPv6 dataplane (SRv6) provides network programming capability to create interoperable overlays with underlay optimization [RFC8986].

This document describes the common failure scenarios and protection mechanisms in SRv6 networks. Then implementation recommendations for protection of SRv6 networks are proposed.

1.1. Requirements Language

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

1.2. Terminology

BE: Best Effort

TE: Traffic Engineering

G-SRv6: Generalized SRv6 Network Programming

2. Forwarding over SRv6 Network

Segment Routing [RFC8402] leverages the source routing paradigm. Segment Routing instantiated on the IPv6 dataplane is referred to as SRv6. SRv6 provides network programming capability to create interoperable overlays with underlay optimization [RFC8986].

In an SRv6 network, the ingress node encapsulates a received packet in an outer IPv6 header, followed by an optional Segment Routing Header (SRH) [RFC8754], which instructs the SRv6 network to forward the packet via a specific path to the egress node. The forwarding path is either an SRv6 BE path or an SRv6 TE path.

2.1. SRv6 BE Path

In the SRv6 BE path, the ingress PE encapsulates the payload in an outer IPv6 header where the destination address is the SRv6 Service SID provided by the egress PE. The underlay P nodes between the PEs only need to perform plain IPv6 shortest path forwarding.

```
            -----------------------
           |      IPv6 Header      |
           |  DA = 2001:DB8:1:1::  |
            -----------------------
           |        Payload        |
            -----------------------
```

Ingress PE ---> P nodes ---> Egress PE

Figure 1: Forwarding over SRv6 BE

2.2. SRv6 TE Path

In the SRv6 TE path, the ingress PE steers the traffic flow into an SR Policy [I-D.ietf-spring-segment-routing-policy], and encapsulates the payload packet in an outer IPv6 header with the Segment Routing Header (SRH) carrying the segment list of the SR policy. The underlay P nodes whose SRv6 SID's are part of the SRH segment list are called endpoint nodes. They will be involved in the forwarding path and execute the function associated with the SID.

```
            -----------------------
           |      IPv6 Header      |
           |  DA = 2001:DB8:6:1::  |
            -----------------------
           |          SRH          |
           | Seg[0]= 2001:DB8:1:1:: |
           | Seg[1]= 2001:DB8:2:1:: |
           | Seg[2]= 2001:DB8:3:1:: |
           | Seg[3]= 2001:DB8:4:1:: |
           | Seg[4]= 2001:DB8:5:1:: |
           | Seg[5]= 2001:DB8:6:1:: |
            -----------------------
           |        Payload        |
            -----------------------
```

Ingress PE ---> P nodes ---> Egress PE

Figure 2: Forwarding over SRv6 TE

If Compressed Segment List encoding is enabled in the SRv6 network [I-D.ietf-spring-srv6-srh-compression], the segment list in the SRH will be encoded in the compressed way. The compressed SRv6 Segment-List encoding can optimize the packet header length by avoiding the repetition of the Locator-Block and trailing bits with each individual SID.

The G-SRv6 mechanism will be used as an example for the encoding of SRv6 TE path in this document. Figure 3 shows the encapsulation of packet using the G-SRv6 mechanism.

```
 ------------------------
|      IPv6 Header       |
|   DA = 2001:DB8:6:1::  |
 ------------------------
|          SRH           |
|Seg[0]= 2001:DB8:1:1::  |
|Seg[1]= 2:1|3:1|4:1|5:1 |
|Seg[2]= 2001:DB8:6:1::  |
 ------------------------
|        Payload         |
 ------------------------
```

Ingress PE ---> P nodes ---> Egress PE

Figure 3: Forwarding over G-SRv6 Encoded TE

3. Protection Mechanisms

3.1. Path Protection

3.1.1. Local Proctection Mechanisms

Local protection is performed by the node adjacent to the failed component using fast-reroute techniques [RFC5286] [RFC5714]. The common method of local repair is to provide a repair path for the destination avoiding the failed component.

[I-D.ietf-rtgwg-segment-routing-ti-lfa] describes the Topology Independent Loop-free Alternate Fast Re-route technology (TI-LFA) using Segment Routing, which is able to provide a loop free backup path irrespective of the topologies used in the network. For each destination in the network, TI-LFA pre-installs a backup forwarding entry for each protected destination ready to be activated upon detection of the failure of a link used to reach the destination.

In SRv6 dataplane, the TI-LFA repair path is encoded as an SRv6 SID list, and encapsulated in the SRH along with an outer IPv6 header. If Compressed Segment List encoding is enabled, the repair node should check the G-SRv6 capability of nodes along the repair path and try to use G-SIDS to encode the repair path, which will help to optimize the packet header length.

3.1.2. Liveness Check For Local Protection

In order to perceive the failures of links and neighbors, a node should monitor the liveness of its adjacent components.

[RFC5880] and [RFC7880] provide widely used mechanisms for liveness check, called Bidirectional Forwarding Detection (BFD) and Seamless Bidirectional Forwarding Detection (S-BFD).

BFD can be associated with the interface state to detect the failure of directly-connected links. Two adjacent nodes may establish BFD or S-BFD sessions between each other, and send BFD control packets to monitor the liveness of each other. In another way, a node may send BFD echo packets to all the neighbors, and they will reflect the packets back, without establishing BFD sessions.

Other OAM methods, such as Ping, TWAMP or STAMP, may also be used for liveness check for local protection, which will not be enumerated here in detail.

3.1.3. Micro-Loop Avoidance

When a component fails or comes back up, the topology is changed. The routing convergence happens in each node at different times and during a different lapse of time. These transient routing inconsistencies may cause micro-loops.

[I-D.bashandy-rtgwg-segment-routing-uloop] provides a mechanism leveraging segment routing to ensure loop-freeness during the IGP reconvergence process, which relies on the temporary use of SR policies ensuring loop-freeness over the post-convergence paths from the converging node to the destination.

In SRv6 dataplane, the loop-free post-convergence path is encoded as an SRv6 SID list, and encapsulated in the SRH along with an outer IPv6 header. If Compressed Segment List encoding is enabled, the converging node should check the G-SRv6 capability of nodes along the post-convergence path and try to use G-SIDs to encode the path.

3.1.4. End-to-End Protection Mechanisms

End-to-end protection lets the ingress PE node be in charge of the failure recovery. The ingress node should steer the flow from the failed path into another alive path.

In the case of SRv6 TE path, the SR Policy itself allows for multiple candidate paths, of which at any point in time there is a single active candidate path that is provisioned in the forwarding plane and used for traffic steering [I-D.ietf-spring-segment-routing-policy]. The candidate path with highest preference is selected as the primary path, and the candidate path with second highest preference can be selected as the hot-standby backup. When the primary candidate path fails, switchover to the backup candidate path can be triggered by fast re-route mechanism.

If all the candidate paths fail, the ingress node may use SRv6 BE path for best-effort forwarding.

3.1.5. Liveness Check For End-to-End Protection

It is essential that the ingress PE node should check the end-to-end liveness of paths, including primary path and backup path. So that the ingress PE node can perceive the path failure and then trigger the switchover.

In the case of SRv6 TE path, BFD or S-BFD can be used to monitor the liveness of SR Policy at the level of segment list. If all the BFD sessions associated with segment lists in a candidate path are down, the candidate path is deemed to be failed. If all the candidate paths is failed, the SR Policy is deemed to be failed.

Moreover, If the SRv6 TE path is strict (every hop along the path appearing in the SID list), the reverse path of the BFD packets should be the same with the forward path. Otherwise, the failure in the reverse path may cause the misjudgement of the liveness of SR Policy. To achieve the consistence of forward path and reverse path, the egress node should be instructed to use specific path to send packets back to the ingress node.

Other OAM methods, such as Ping, TWAMP or STAMP, may also be used for liveness check for end-to-end protection, which will not be enumerated here in detail.

Local protection and end-to-end protection may both be used in the same SRv6 network. Since the speed of failure detection for local protection is faster than end-to-end protection, local protection usually performs the local repair in advance, which allows the path to remain alive. In this case, the ingress node will not perceive the failure and does not need to trigger end-to-end protection.

3.2. Service Protection

If the failure occurs on the egress PE node, the service provided by that PE is not accessible anymore. TI-LFA or the hot-standby backup candidate path of SR Policy will not work under this circumstance. To provide protection, the packet should be forwarded to another backup Egress PE node of the same service, if it exists.

3.2.1. Local Repair

In the case of egress PE node failure, the local repair node should forward packet to another Egress PE node.

[I-D.ietf-rtgwg-srv6-egress-protection] provides a method to use Mirror SID for egress protection. The Mirror SID is configured on the backup egress PE to protect the primary egress PE, and it will be used by the repair node to encode the segment list of repair path.

If a failure occurs on the link between egress PE and CE, that PE should work as the local repair node and forward packet to another Egress PE node. It can be achieved by using the tunnel towards another PE as an alternate next-hop for the routes to CE with FRR mechanism, while the failed link is the primary next-hop. The alternate next-hop can be installed as an SRv6 BE path when receiving the BGP routes of the same service from another egress PE.

3.2.2. Ingress Node Switchover

If there are multiple egress PE nodes, the ingress PE node receives all their advertisements of the same service, and builds paths for each of them respectively. The ingress PE node may use Fast Reroute (FRR) for these different paths. When the primary egress PE node fails, the ingress node steers the flow to the path belonging to another egress PE node for protection.

BFD can be used to monitor the liveness of the service SID, locator or interface address of the egress PE node. If the BFD session is down, the egress PE node is deemed to be unreachable.

Service protection and path protection may both be used in the same SRv6 network. Among the different paths to the same egress PE node and the paths to different egress PE nodes, one is selected as the primary path and others are used as backup. The priorities of multiple backup paths may be decided by the egress-node-first strategy or the TE-first strategy.

By the Egress-node-first strategy, paths to the primary egress PE nodes are prioritized. For example, if a failure occurs on the primary path, the ingress PE node will select another path still leading to the primary egress PE nodes. Unless all the paths to the primary egress PE node are failed, the ingress PE node would use the path to the backup egress PE node.

By the TE-first strategy, SRv6 TE paths to any egress PE node have higher priorities than SRv6 BE paths. For example, if a failure occurs on the primary path and there is no other alive SRv6 TE paths to the primary egress PE node, the ingress node will select an SRv6 TE path to the backup egress PE node, rather than an SRv6 BE path still leading to the primary egress PE node.

4. Implementation Recommendations

This section will introduce the implementation recommendations of protection for SRv6 BE and SRv6 TE scenarios in SRv6 network:

Figure 5 is used as a reference topology in this section. PE1 and PE3 are primary PE nodes for VPN service access. PE2 and PE4 are used as backup. The prefix of CE2, along with VPN service SID, is advertised by BGP routes from PE3 and PE4 to PE1 and PE2. The VPN traffic is from CE1 to CE2.

```
     PE1-----P1-----P3-----P5-----P7----PE3
    / | \  / | \  / | \  / | \  / | \  / | \
   /  |  \/  |  \/  |  \/  |  \/  |  \/  |  \
 CE1  |  /\  |  /\  |  /\  |  /\  |  /\  |  CE2
   \  | /  \ | /  \ | /  \ | /  \ | /  \ |  /
    \ |/    \|/    \|/    \|/    \|/    \| /
     PE2-----P2-----P4-----P6-----P8----PE4
```

Figure 5: Reference Topology

The link metrics are configured as follows:

o Metrics of PE1-P2, PE2-P1, P1-P4, P2-P3, P3-P6, P4-P5, P5-P8, P6-P7, P7-PE4, P8-PE3, PE1-PE2 and PE3-PE4 links are 11.

o Metrics of all other links are 5.

o Link metrics are bidirectional.

All P and PE nodes are capable of G-SRv6 compression. The SRv6 SIDs are configured using the following rules.

```
NodeID:                 An for PEn, Bn for Pn
Locator:                2001:DB8:NodeID::/48
End SID:                Locator:1::
End SID with COC:       Locator:2::
End DT:                 Locator:100:: (Only for PE nodes)
End.X SID:              Locator:NeigborNodeID + F1::
End.X SID with COC:     Locator:NeigborNodeID + F2::
```

For example, the SRv6 SIDs configured for PE1 and P8 are as follows.

```
PE1:
  Locator:              2001:DB8:A1::/48
  End SID:              2001:DB8:A1:1::
  End SID with COC:     2001:DB8:A1:2::
  End DT:               2001:DB8:A1:100::
  For PE1->P1:
    End.X SID:          2001:DB8:A1:B1F1::
    End.X SID with COC: 2001:DB8:A1:B1F2::
  For PE1->P2:
    End.X SID:         2001:DB8:A1:B2F1::
    End.X SID with COC: 2001:DB8:A1:B2F2::

P8:
  Locator:              2001:DB8:B8::/48
  End SID:              2001:DB8:B8:1::
  End SID with COC:     2001:DB8:B8:2::
  For P8->P5:
    End.X SID:          2001:DB8:B8:B5F1::
    End.X SID with COC: 2001:DB8:B8:B5F2::
  For P8->P6:
    End.X SID:         2001:DB8:B8:B6F1::
    End.X SID with COC: 2001:DB8:B8:B6F2::
  For P8->P7:
    End.X SID:         2001:DB8:B8:B7F1::
    End.X SID with COC: 2001:DB8:B8:B7F2::
  For P8->PE3:
    End.X SID:         2001:DB8:B8:A3F1::
    End.X SID with COC: 2001:DB8:B8:A3F2::
  For P8->PE4:
    End.X SID:         2001:DB8:B8:A4F1::
    End.X SID with COC: 2001:DB8:B8:A4F2::
```

The SR Policies on PE1 are configured as follows:

```
SR Policy 1 (Strict Path to PE3)
  Candidate Path 1
    Preference: 20
    Segment List: 2001:DB8:A1:B1F2::, 2001:DB8:B1:B3F2::,
2001:DB8:B3:B5F2::, 2001:DB8:B5:B7F2::, 2001:DB8:B7:A3F1::
  Candidate Path 2
    Preference: 10
    Segment List: 2001:DB8:A1:B2F2::, 2001:DB8:B2:B4F2::,
2001:DB8:B4:B6F2::, 2001:DB8:B6:B8F2::,2001:DB8:B8:A3F1::

SR Policy 2 (Loose Path to PE4)
  Candidate Path 1
    Preference: 20
    Segment List: 2001:DB8:B4:2::, 2001:DB8:B8:2::,2001:DB8:A4:1::
```

4.1. SRv6 BE

In this scenario, SRv6 BE paths are used to steer the VPN service. The deployments of protection are as follows:

- o All nodes enable TI-LFA for local protection.

- o All nodes enable BFD for links and neighbors.

- o Ingress PE node enables FRR of SRv6 BE path to backup egress PE node for service protection.

- o Ingress PE node enables BFD for locator of egress PE node to monitor the liveness of SRv6 BE path.

PE1 installs the SRv6 BE path to PE3 with destination address 2001:DB8:A3:100:: as the primary next-hop for the VPN flow. Meanwhile, PE1 also installs the SRv6 BE path to PE4 with destination address 2001:DB8:A4:100:: as the backup next-hop.

PE1 enables BFD for locator 2001:DB8:A3::/48 and 2001:DB8:A4::/48 to monitor the liveness of SRv6 BE paths.

TI-LFA is enabled on all nodes. Take P1 for example. The shortest path from P1 to PE3 is via neighbor P3. In order to provide local protection for P3 node failure, P1 computes and installs the repair path P1->P2->P4->P6, using [2001:DB8:B4:2::, B4:B6F1] as the G-SRv6 SID list.

All nodes use BFD to monitor the liveness of links and adjacent nodes.

Under normal circumstances, PE1 encapsulates the VPN payload in an outer IPv6 header where the destination address is 2001:DB8:A3:100::.

Assume that a failure occurs on P3. The fail-timer of BFD from P1 to P3 expires, so P1 perceives the failure. When P1 forwards the VPN packet, the TI-LFA repair path is used. Then, P1 encapsulates the packet in an outer IPv6 Header with SRH carrying a compressed segment-list of [2001:DB8:B4:2::, B4:B6F1], as shown in the following figure. The packet is forwarded in the repair path P1->P2->P4->P6 according to the outer IPv6 Header and SRH. So the failure is repaired by local protection.

```
                                 ------------------------
                                |      IPv6 Header       |
                                |  DA = 2001:DB8:B4:2::  |
                                 ------------------------
                                |          SRH           |
                                |Seg[0]= B4:B6F1         |
                                |Seg[1]= 2001:DB8:B4:2:: |
------------------------       ------------------------
|      IPv6 Header       |     |      IPv6 Header       |
| DA = 2001:DB8:A3:100:: |     | DA = 2001:DB8:A3:100:: |
 ------------------------       ------------------------
|      VPN Payload       |     |      VPN Payload       |
 ------------------------       ------------------------

PE1         --->           P1            --->         P2
```

Assume that a failure occurs on PE3. TI-LFA does not work and the packets along the SRv6 BE path are dropped. Then the BFD session from PE1 to locator 2001:DB8:A3::/48 is down, so PE1 triggers the switchover to the SRv6 BE path to PE4 and encapsulates the VPN payload in an outer IPv6 header where the destination address is 2001:DB8:A4:100::. After that, the VPN traffic from CE1 to CE2 is recovered.

Assume that a failure occurs on link PE3-CE2. Since the BFD session from PE1 to locator 2001:DB8:A3::/48 is still alive, PE1 continues to forward the VPN packets to PE3. When PE3 receives the packet, it removes the outer IPv6 header, looks up the VPN table and forwards the packet to CE2. However, the link PE3-CE2 is failed. So PE3 selects the FRR alternate next-hop which is the SRv6 BE path to PE4. Then PE3 encapsulates the packet in an outer IPv6 header where the destination address is 2001:DB8:A4:100::, and forwards it through the link PE3-PE4.

4.2. SRv6 TE

In this scenario, the SRv6 TE strict path with G-SRv6 compression is used to steer the VPN traffic flows to the primary egress node PE3, and the SRv6 TE loose path with G-SRv6 compression is used for the backup egress node PE4.

The deployments of protection are as follows:

- o In the SR Policy of SRv6 TE strict path, disjoint backup candidate path is used as hot standby for end-to-end protection.
- o Ingress PE node uses SRv6 BE paths as backup for end-to-end protection of SRv6 TE paths.
- o Ingress PE node enables BFD for SR Policy. In the case of SRv6 TE strict path, the reverse path of BFD packet keeps consistent with forward path.
- o Ingress PE node enables BFD for locator of egress PE node to monitor the liveness of SRv6 BE path.
- o Ingress PE node enables FRR of paths to backup egress PE node for service protection.
- o All nodes enable TI-LFA for local protection. All nodes enable BFD for links and neighbors.

PE1 installs SR Policy 1, which is the SRv6 TE strict path to PE3, as the primary next-hop for the VPN flow. SR Policy 1 has two disjoint candidate paths. The candidate path with higher preference is selected as the primary candidate path, and the candidate path with lower preference is selected as hot standby backup.

Meanwhile, the SRv6 BE path to PE3, the SRv6 TE loose path to PE4 (SR Policy 2), and the SRv6 BE path to PE4 are also installed as backup next-hops. The priorities of multiple backup paths may be decided by either of the egress-node-first strategy or the TE-first strategy.

Egress-node-first strategy:

- o primary: SRv6 TE path to primary egress node PE3 (SR Policy 1)
- o backup(1st priority): SRv6 BE path to primary egress node PE3

o backup(2nd priority): SRv6 TE path to backup egress node PE4 (SR Policy 2)

o backup(3rd priority): SRv6 BE path to backup egress node PE4

TE-first strategy:

o primary: SRv6 TE path to primary egress node PE3 (SR Policy 1)

o backup(1st priority): SRv6 TE path to backup egress node PE4 (SR Policy 2)

o backup(2nd priority): SRv6 BE path to primary egress node PE3

o backup(3rd priority): SRv6 BE path to backup egress node PE4

Egress-node-first strategy is used as an example below.

PE1 enables BFD for SR Policy 1 and SR Policy 2 to monitor the liveness of SRv6 TE paths. For SR Policy 1 which is the strict path, the forward and reverse paths of BFD packet should be the same. For example, the primary path of SR Policy 1 is PE1->P1->P3->P5->P7->PE3, so the reverse path should be PE3->P7->P5->P3->P1->PE1. A segment list of such reverse path is installed on PE3. PE1 may send BFD packet with the segment list of SR Policy 1 along with the BSID of reverse path. When the BFD packet is forwarded along the strict path to PE3, PE3 will add an outer IPv6 header with SRH carrying the segment list of [2001:DB8:A3:B7F2::, B7:B5F2, B5:B3F2, B3:B1F2, B1:A1F1], which instructs the packet to be forwarded along the same strict path back to PE1.

PE1 enables BFD for locator 2001:DB8:A3::/48 and 2001:DB8:A4::/48 to monitor the liveness of SRv6 BE paths.

TI-LFA is enabled on all nodes. BFD are used to monitor the liveness of links and adjacent nodes.

Under normal circumstances, PE1 encapsulates the VPN payload in an outer IPv6 header with SRH carrying the segment list of primary candidate path of SR Policy 1 along with the VPN SID advertised by PE3. Using G-SRv6 compression, the segment list will be encoded as [2001:DB8:A1:B1F2::, B1:B3F2, B3:B5F2, B5:B7F2, B7:A3F1, 2001:DB8:A3:100::].

Assume that a failure occurs on P3. The packets are dropped since the failed P3 is on the path. The BFD session of the segment list in

the primary candidate path of SR Policy 1 is down, so PE1 triggers the switchover to the backup candidate path of SR Policy 1. Then PE1 encapsulates the VPN payload in an outer IPv6 header with SRH carrying the segment list of [2001:DB8:A1:B2F2::, B2:B4F2, B4:B6F2, B6:B8F2, B8:A3F1, 2001:DB8:A3:100::].

Before the recovery of P3, assume that P8 also fails. The BFD session of the segment list in the backup candidate path of SR Policy 1 is also down. Then PE1 triggers the switchover to the 1st priority backup next-hop which is the SRv6 BE path to PE3. PE1 encapsulates the VPN payload in an outer IPv6 header where the destination address is 2001:DB8:A3:100::.

Assume that a failure occurs on PE3. Both the BFD sessions of SR Policy 1 and locator 2001:DB8:A3::/48 are down, which means the primary next-hop and the 1st priority backup next-hop are down. So PE1 triggers the switchover to the 2nd priority backup next-hop, which is the SRv6 TE loose path to PE4. Then PE1 encapsulates the VPN payload in an outer IPv6 header with SRH carrying the segment list of [2001:DB8:B4:2::, B8:2, A4:1, 2001:DB8:A4:100::].

Before the recovery of PE3, assume that a failure occurs on P6. The fail-timer of BFD from P4 to P6 expires, so P4 perceives the failure. When P4 forwards the VPN packet, the TI-LFA repair path is used. Then, P4 encapsulates the packet in an outer IPv6 Header with SRH carrying a compressed segment-list of [2001:DB8:B5:2::, B5:B7F1]. The packet is forwarded in the repair path P4->P3->P5->P7 according to the outer IPv6 Header and SRH. So the failure is repaired by local protection.

Before the recovery of PE3, assume that a failure occurs on P8. When P6 forwards the VPN packet to destination address 2001:DB8:B8:2:: which is one of the segments in the segment list of SRH, the TI-LFA on P6 does not work, since the failed node P8 is the destination. So the packets are dropped. The BFD session of SR Policy 2 is down, and PE1 triggers the switchover to the 3rd priority backup next-hop which is the SRv6 BE path to PE4. Then PE1 encapsulates the VPN payload in an outer IPv6 header where the destination address is 2001:DB8:A4:100::. If the routing convergence is not completed at the moment, P6 will use TI-LFA repair path P6->P5->P7->PE4 to forward the packet. After the routing convergence is done, P nodes will forward the packet along new shortest path excluding P8.

Assume that a failure occurs on link PE3-CE2. This is similar with the same failure in section 4.1. The BFD session is still alive, PE1 continues to forward the VPN packets to PE3. PE3 will select the FRR

alternate next-hop for CE1 and forward the packet to PE4 with SRv6 BE path.

5. Security Considerations

TBD.

6. IANA Considerations

This document has no IANA actions.

7. Contributors

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

Mengxiao Chen
H3C
Email: chen.mengxiao@h3c.com

8. References

8.1. Normative References

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

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, May 2017

[RFC8754] Filsfils, C., Ed., Dukes, D., Ed., Previdi, S., Leddy, J., Matsushima, S., and D. Voyer, "IPv6 Segment Routing Header (SRH)", RFC 8754, DOI 10.17487/RFC8754, March 2020, <https://www.rfc-editor.org/info/rfc8754>.

[I-D.ietf-spring-segment-routing-policy] Filsfils, C., Talaulikar, K., Voyer, D., Bogdanov, A., and P. Mattes, "Segment Routing Policy Architecture", draft-ietf-spring-segment-routing-policy-22 (work in progress), March 2022.

[I-D.ietf-spring-srv6-srh-compression] Cheng, W., Filsfils, C., Li, Z., Decraene, B., Cai, D., Clad, F., Zadok, S., Guichard, J., Aihua, L., Raszuk, R. and C. Li, " Compressed SRv6 Segment List Encoding in SRH", draft-ietf-spring-srv6-srh-compression-01 (work in progress), March 2022.

[I-D.ietf-rtgwg-segment-routing-ti-lfa] Litkowski, S., Bashandy, A., Filsfils, C., Francois, P., Decraene, B., and D. Voyer, "Topology Independent Fast Reroute using Segment Routing", draft-ietf-rtgwg-segment-routing-ti-lfa-08 (work in progress), January 2022.

[RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD)", RFC 5880, DOI 10.17487/RFC5880, June 2010, <https://www.rfc-editor.org/info/rfc5880>.

[RFC7880] Pignataro, C., Ward, D., Akiya, N., Bhatia, M., and S. Pallagatti, "Seamless Bidirectional Forwarding Detection (S-BFD)", RFC 7880, DOI 10.17487/RFC7880, July 2016, <https://www.rfc-editor.org/info/rfc7880>.

8.2. Informative References

[RFC8402] Filsfils, C., Ed., Previdi, S., Ed., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", RFC 8402, DOI 10.17487/RFC8402, July 2018, <https://www.rfc-editor.org/info/rfc8402>.

[RFC8986] Filsfils, C., Ed., Camarillo, P., Ed., Leddy, J., Voyer, D., Matsushima, S., and Z. Li, "Segment Routing over IPv6 (SRv6) Network Programming", RFC 8986, DOI 10.17487/RFC8986, February 2021, <https://www.rfc-editor.org/info/rfc8986>.

[RFC5286] Atlas, A., Ed. and A. Zinin, Ed., "Basic Specification for IP Fast Reroute: Loop-Free Alternates", RFC 5286, DOI 10.17487/RFC5286, September 2008, <https://www.rfc-editor.org/info/rfc5286>.

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

[I-D.bashandy-rtgwg-segment-routing-uloop] Bashandy, A., Filsfils, C., Litkowski, S., Decraene, B., Francois, P. and P., Psenak, "Loop avoidance using Segment Routing", draft-bashandy-rtgwg-segment-routing-uloop-13 (work in progress), June 2022.

[I-D.ietf-rtgwg-srv6-egress-protection] Hu, Z., Chen, H., Chen, H., Wu, P., Toy, M., Cao, C., He, T., Liu, L., and X. Liu, "SRv6 Path Egress Protection", Work in Progress, Internet-Draft, draft-ietf-rtgwg-srv6-egress-protection-05, April 2022.

Authors' Addresses

Yisong Liu
China Mobile
China

Email: liuyisong@chinamobile.com

Weiqiang Cheng
China Mobile
China

Email: chengweiqiang@chinamobile.com

Changwang Lin
New H3C Technologies
China

Email: linchangwang.04414@h3c.com

Xuesong Geng
Huawei Technologies
China

Email: gengxuesong@huawei.com

Yao Liu
ZTE Corp.
China

Email: liu.yao71@zte.com.cn

Network Working Group
Internet-Draft
Intended status: Experimental
Expires: September 8, 2022

R. Miao
H. Liu
Alibaba Group
R. Pan
J. Lee
C. Kim
Intel Corporation
B. Gafni
Y. Shpigelman
Mellanox Technologies, Inc.
J. Tantsura
Microsoft Corporation
March 7, 2021

HPCC++: Enhanced High Precision Congestion Control
draft-miao-rtgwg-hpccplus-00

Abstract

Congestion control (CC) is the key to achieving ultra-low latency, high bandwidth and network stability in high-speed networks. However, the existing high-speed CC schemes have inherent limitations for reaching these goals.

In this document, we describe HPCC++ (High Precision Congestion Control), a new high-speed CC mechanism which achieves the three goals simultaneously. HPCC++ leverages inband telemetry to obtain precise link load information and controls traffic precisely. By addressing challenges such as delayed signaling during congestion and overreaction to the congestion signaling using inband and granular telemetry, HPCC++ can quickly converge to utilize all the available bandwidth while avoiding congestion, and can maintain near-zero in-network queues for ultra-low latency. HPCC++ is also fair and easy to deploy in hardware, implementable with commodity NICs and switches.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

The link speed in data center networks has grown from 1Gbps to 100Gbps in the past decade, and this growth is continuing. Ultralow latency and high bandwidth, which are demanded by more and more applications, are two critical requirements in today's and future high-speed networks.

Given that traditional software-based network stacks in hosts can no longer sustain the critical latency and bandwidth requirements as described in [Zhu-SIGCOMM2015], offloading network stacks into hardware is an inevitable direction in high-speed networks. As an example, large-scale networks with RDMA (remote direct memory access) often uses hardware-offloading solutions. In some cases, the RDMA networks still face fundamental challenges to reconcile low latency, high bandwidth utilization, and high stability.

This document describes a new congestion control mechanism, HPCC++ (Enhanced High Precision Congestion Control), for large-scale, high-speed networks. The key idea behind HPCC++ is to leverage the precise link load information from signaled through inband telemetry to compute accurate flow rate updates. Unlike existing approaches that often require a large number of iterations to find the proper flow rates, HPCC++ requires only one rate update step in most cases. Using precise information from inband telemetry enables HPCC++ to address the limitations in current congestion control schemes. First, HPCC++ senders can quickly ramp up flow rates for high utilization and ramp down flow rates for congestion avoidance. Second, HPCC++ senders can quickly adjust the flow rates to keep each link's output rate slightly lower than the link's capacity, preventing queues from being built-up as well as preserving high link utilization. Finally, since sending rates are computed precisely based on direct measurements at switches, HPCC++ requires merely three independent parameters that are used to tune fairness and efficiency.

The base form of HPCC++ is the original HPCC algorithm and its full description can be found in [SIGCOMM-HPCC]. While the original

design lays the foundation for inband telemetry based precision congestion control, HPCC++ is an enhanced version which takes into account system constraints and aims to reduce the design overhead and further improves the performance. Section 6 describes these detailed proposed design enhancements and guidelines.

This document describes the architecture changes in switches and end-hosts to support the needed tranmission of inband telemetry and its consumption, that imporves the efficiency in handling network congestion.

2. Terminology

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

3. System Overview

Figure 1 shows the end-to-end system that HPCC++ operates in. During the traverse of the packet from the sender to the receiver, each switch along the path inserts inband telemetry that reports the current state of the packet’s egress port, including timestamp (ts), queue length (qLen), transmitted bytes (txBytes), and the link bandwidth capacity (B), together with switch_ID and port_ID. When the receiver gets the packet, it may copy all the inband telemetry recorded from the network to the ACK message it sends back to the sender, and then the sender decides how to adjust its flow rate each time it receives an ACK with network load information. Alternatively, the receiver may calculate the flow rate based on the inband telemetry information and feedback the calculated rate back to the sender. The notification packets would include delayed ack information as well.

Note that there also exist network nodes along the reverse (potentially uncongested) path that the feedback reports traverse. Those network nodes are not shown in the figure for sake of brevity.

```
+---------+  pkt    +-------+ pkt+tlm +-------+ pkt+tlm +----------+
|  Data   |-------->|       |-------->|       |-------->| Data     |
|  Sender |=========|Switch1|=========|Switch2|=========| Receiver |
+---------+ Link-0  +-------+  Link-1 +-------+  Link-2 +----------+
    /|\                                                        |
     |                                                        |
     +--------------------------------------------------------+
                    Notification Packets/ACKs
```

Figure 1: System Overview (tlm=inband telemtry)

- o Data sender: responsible for controlling inflight bytes. HPCC++ is a window-based congestion control scheme that controls the number of inflight bytes. The inflight bytes mean the amount of data that have been sent, but not acknowledged by the sender yet. Controlling inflight bytes has an important advantage compared to controlling rates. In the absence of congestion, the inflight bytes and rate are interchangeable with equation inflight = rate * T where T is the base propagation RTT. The rate can be calculated locally or obtained from the notification packet. The sender may further use the data pacing mechanism, potentially implemented in hardware, to limit the rate accordingly.

- o Network nodes: responsible of inserting the inband telemetry information to the data packet. The inband telemetry information reports the current load of the packet's egress port, including timestamp (ts), queue length (qLen), transmitted bytes (txBytes), and link bandwidth capacity (B). Besides, the inband telemetry contains switch_ID and port_ID to identify a link.

- o Data receiver: responsible for either reflecting back the inband telemetry information in the data packet or calculating the proper flow rate based on network congestion information in inband telemetry and sending notification packets back to the sender.

4. HPCC++ Algorithm

HPCC++ is a window-based congestion control algorithm. The key design choice of HPCC++ is to rely on network nodes to provide fine-grained load information, such as queue size and accumulated tx/rx traffic to compute precise flow rates. This has two major benefits: (i) HPCC++ can quickly converge to proper flow rates to highly utilize bandwidth while avoiding congestion; and (ii) HPCC++ can consistently maintain a close-to-zero queue for low latency.

This section introduces the list of notations and describes the core congestion control algorithm.

4.1. Notations

This section summarizes the list of variables and parameters used in the HPCC++ algorithm. Figure 3 also includes the default values for choosing the algorithm parameters either to represent a typical setting in practical applications or based on theoretical and simulation studies.

Notation	Variable Name
W_i	Window for flow i
Wc_i	Reference window for flow i
B_j	Bandwidth for Link j
I_j	Estimated inflight bytes for Link j
U_j	Normalized inflight bytes for Link j
qlen	Telemetry info: link j queue length
txRate	Telemetry info: link j output rate
ts	Telemetry info: timestamp
txBytes	Telemetry info: link j total transmitted bytes associated with timestamp ts

Figure 2: List of variables.

Notation	Parameter Name	Default Value
T	Known baseline RTT	5us
eta	Target link utilization	95%
maxStage	Maximum stages for additive increases	5
N	Maximum number of flows	...
W_ai	Additive increase amount	...

Figure 3: List of algorithm parameters and their default values.

4.2. Design Functions and Procedures

The HPCC++ algorithm can be outlined as below:

```
1: Function MeasureInflight(ack)
2:    u = 0;
3:    for each link i on the path do
4:                   ack.L[i].txBytes-L[i].txBytes
          txRate =  ----------------------------- ;
                         ack.L[i].ts-L[i].ts
5:               min(ack.L[i].qlen,L[i].qlen)      txRate
           u' = ---------------------------- +  ---------- ;
                     ack.L[i].B*T                ack.L[i].B
6:         if u' > u then
7:             u = u'; tau = ack.L[i].ts -  L[i].ts;
8:     tau = min(tau, T);
9:     U = (1 - tau/T)*U + tau/T*u;
10:    return U;

11: Function ComputeWind(U, updateWc)
12:    if U >= eta or incStage >= maxStagee then
13:             Wc
           W = ----- + W_ai;
               U/eta
14:        if updateWc then
15:            incStagee = 0; Wc = W ;
16:    else
17:        W = Wc + W_ai ;
18:        if updateWc then
19:            incStage++; Wc = W ;
20:    return W

21: Procedure NewAck(ack)
22:    if ack.seq > lastUpdateSeq then
23:        W = ComputeWind(MeasureInflight(ack), True);
24:        lastUpdateSeq = snd_nxt;
25:    else
26:        W = ComputeWind(MeasureInflight(ack), False);
27:    R = W/T; L = ack.L;
```

The above illustrates the overall process of CC at the sender side for a single flow. Each newly received ACK message triggers the procedure NewACK at Line 21. At Line 22, the variable lastUpdateSeq is used to remember the first packet sent with a new W c , and the sequence number in the incoming ACK should be larger than lastUpdateSeq to trigger a new sync betweenW c andW (Line 14-15 and 18-19). The sender also remembers the pacing rate and current inband telemetry information at Line 27. The sender computes a new window size W at Line 23 or Line 26, depending on whether to update W c , with function MeasureInflight and ComputeWind. Function MeasureInflight estimates normalized inflight bytes with Eqn (2) at Line 5. First, it computes txRate of each link from the current and

last accumulated transferred bytes txBytes and timestamp ts (Line 4). It also uses the minimum of the current and last qlen to filter out noises in qlen (Line 5). The loop from Line 3 to 7 selects maxi(Ui) in Eqn. (3). Instead of directly using maxi(Ui), we use an EWMA (Exponentially Weighted Moving Average) to filter the noises from timer inaccuracy and transient queues. (Line 9). Function ComputeWind combines multiplicative increase/ decrease (MI/MD) and additive increase (AI) to balance the reaction speed and fairness. If a sender finds it should increase the window size, it first tries AI for maxStage times with the stepWAI (Line 17). If it still finds room to increase after maxStage times of AI or the normalized inflight bytes is above, it calls Eqn (4) once to quickly ramp up or ramp down the window size (Line 12-13).

5. Configuration Parameters

HPCC++ has three easy-to-set parameters: eta, maxStagee, and W_ai. eta controls a simple tradeoff between utilization and transient queue length (due to the temporary collision of packets caused by their random arrivals, so we set it to 95% by default, which only loses 5% bandwidth but achieves almost zero queue. maxStage controls a simple tradeoff between steady state stability and the speed to reclaim free bandwidth. We find maxStage = 5 is conservatively large for stability, while the speed of reclaiming free bandwidth is still much faster than traditional additive increase, especially in high bandwidth networks. W_ai controls the tradeoff between the maximum number of concurrent flows on a link that can sustain near-zero queues and the speed of convergence to fairness. Note that none of the three parameters are reliability-critical.

HPCC++’s design brings advantages to short-lived flows, by allowing flows starting at line-rate and the separation of utilization convergence and fairness convergence. HPCC++ achieves fast utilization convergence to mitigate congestion in almost one round-trip time, while allows flows to gradually converge to fairness. This design feature of HPCC++ is especially helpful for the workload of datacenter applications, where flows are usually short and latency-sensitive. Normally we set a very small W_ai to support a large number of concurrent flows on a link, because slower fairness is not critical. A rule of thumb is to set W_ai = W_init*(1-eta) / N where N is the expected or receiver reported maximum number of concurrent flows on a link. The intuition is that the total additive increase every round (N*W_ai) should not exceed the bandwidth headroom, and thus no queue forms. Even if the actual number of concurrent flows on a link exceeds N, the CC is still stable and achieves full utilization, but just cannot maintain zero queues.

6. Design enhancement and implementation

There are three compoments HPCC++ needs to implement: telementry padding, congestion notification, and rate update.

6.1. Inband telemetry padding at the network switches

HPCC++ only relies on packets to share information across senders, receivers, and switches. The switch should capture inband telemetry information that includes link load (txBytes, qlen, ts) and link spec (switch_ID, port_ID, B) at the egress port. Note, each switch should record all those information at the single snapshot to achieve a precise link load estimate. Inside a data center, the path length is often no more than 5 hops. The overhead of the inband telemetry padding for HPCC++ is considered to be low.

As long the above algorithm is met, HPCC++ is open to a variety of inband telemetry format standards, which are orthogonal to the HPCC++ algorithm. Although this document does not mandate a particular inband telemetry header format or encapsulation, we provide concrete implementation specifications using strandard inband telemetry protocols, including IFA [I-D.ietf-kumar-ippm-ifa], IETF IOAM [I-D.ietf-ippm-ioam-data], and P4.org INT [P4-INT]. In fact, the emerging inband telemetry protocols inform the evolution for a broader range of protocols and network functions, where this document leverages the trend to propose the architecture change to support in-network functions like congestion control with high efficiency.

6.1.1. Inband telemetry on IFA2.0

For more details, please refer to IFA [I-D.ietf-kumar-ippm-ifa]

6.1.2. Inband telemetry on IOAM

Please refer to IETF IOAM [I-D.ietf-ippm-ioam-data]

6.1.3. Inband telemetry on P4

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  nHop |        pathID         |          Padding              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Speed |                    Timestamp                  |txBytes|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      txBytes(lower)           |         Queue Length          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            2nd Hop                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            2nd Hop(lower)                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Options                    |    Padding    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 4: Example P4.org INT header

Figure 4 shows the packet format of the INT padding after UDP header. The field nHop is the hop count of the packet's path. The field pathID is the XOR of all the switch IDs (which are 12 bits) along the path. The sender sets nHop and pathID to 0. Each switch along the path adds nHop by 1, and XORs its own switch ID to the pathID. The sender uses pathID to judge whether the path of the flow has been changed. If so, it throws away the existing status records of the flow and builds up new records. Each switch has an 8-byte field to record the status of the egress port of the packet when the packet is emitted. B is a enum type which indicates the speed type of the port(e.g. 40Gbps, 100Gbps, etc.). Timestamp (24 bits) is when the packet is emitted from its egress port, txBytes (20 bits) is the accumulative total bytes sent from the egress port, and Queue length (16 bits) is the current queue length of the egress port.

6.2. Congestion Notification

HPCC++ uses congestion notification to fetch network congestion information from switches for proper rate updates at end-hosts. Although the basic algorithm described in Section 4 is to add inband telemetry information into every data packet for optimal performance, HPCC++ supports flexible implementation choices to work seamly with transport protocol stacks. We consider congestion nofication choices in both forward and reverse directions of the traffic.

6.2.1. Forward direction Congestion detection

Forward direction is the traffic direction of data packets that experience bandwidth contention and possible network congestion. The function of congestion notification in forward direction is to fetch inband telemetry from switches. HPCC++ defines two approaches of doing this.

1. Inband with data packet.

This is basic algorithm setting described in Section 4, where the end-host inserts inband telemetry header into data packets. Switches along the path detect the inband telemetry header and correspondingly add inband telemetry information into data packet to react to congestion as soon as the very first packet observing the network congestion. This is especially helpful to reduce the risk of severe congestion in incast scenarios at the first round-trip time. In addition, original HPCC's algorithm introduction of Wc is for the purpose of solving the over-reaction issue from using this per-packet response. Different with in Section 4, end-host can choice uses every data packet or only a subset of data packets to reduce the overhead. To insert telemetry header, differet telemetry protocols have specific settings for IFA, IETF IOAM, and P4.org INT as following.

2. Probe packet.

Switches touching every data packet for inband telemetry inserting may lead to security or performance concerns, HPCC++ supports the ``out-of-band'' approach that uses special-generated probe packets at end-hosts to fetch inband telemetry from switches. Thereby, the probe packets should take the same routing path and QoS queueing with the data packets. End-hosts can generate probe packets less frequently and we recommend once per round trip time. In addition, the end-host issues probe packets only when it has data packet in the flight.

6.2.2. Reverse direction

Reverse direction is the receiver conveying inband telemetry back to traffic sender for rate updates. Similar to forward direction, there are also inband and out-of-band approaches.

1. Inband with ACK packet.

HPCC++ supports to use the ACK packet in transport protocols to convey the inband telemetry. TCP generates ACK packet once per every

data packet or per a few data packets. With ACK packet, the receive sends accumulated inband telemetry back to sender for rate updates.

2. Notification packet.

Using ACK packet for inband telemetry notification requires transport stack modification and sometimes leads to delay in notification when certain delayed acknowledged mechanism is used. Hence, HPCC++ allows the receiver to use special-generated notification packets to deliver inband telemetry. The nofication packet is generated per each probe packet or data packet with inband telemetry.

6.3. Congestion control at NICs

6.3.1. Sender-based HPCC

Figure 5 shows HPCC++ implementation on a NIC. The NIC provides an HPCC++ module that resides on the data path of the NIC, HPCC++ modules realize both sender and receiver roles.

```
+-------------------------------------------------------------------+
|  +---------+ window update +-----------+ PktSend +-----------+   |
|  |         |-------------->| Scheduler |-------> |Tx pipeline|---+->
|  |         | rate update   +-----------+         +-----------+   |
|  |  HPCC++ |                                           ^         |
|  |         |                            inband telemetry|         |
|  |  module |                                           |         |
|  |         |                                     +-----+-----+   |
|  |         |<----------------------------------- |Rx pipeline| <-+--
|  +---------+      telemetry response event        +-----------+   |
+-------------------------------------------------------------------+
```

Figure 5: Overview of NIC Implementation

1. Sender side flow

The HPCC++ module running the HPCC CC algorithm in the sender side for every flow in the NIC. Flow can be defined by some transport parameters including 5-tuples, destination QP (queue pair), etc. It receives inband telemetry response events per flow which are generated from the RX pipeline, adjusts the sending window and rate, and update the scheduler on the rate and window of the flow.

The scheduler contains a pacing mechanism that determine the flow rate by the value it got from the algorithm. It also maintains the current sending window size for active flows. If the pacing

mechanism and the flow's sending window permits, the scheduler invokes for the flow a PktSend command to TX pipeline.

The TX pipeline implements packet processing. Once it receives the PktSend event with flow ID from the scheduler, it generates the corresponding packet and delivers to the Network. If a sent packet should collect telemetry on its way the TX pipeline may add indications/headers that triggers the network elements to add telemetry data according to the inband telemetry protocol in use. The telemetry can be collected by the data packet or by dedicated prob packets generated in the TX pipeline.

The RX pipe parses the incoming packets from the network and identifies whether telemetry is embedded in the parsed packet. On receiving a telemetry response packet, the RX pipeline extracts the network status from the packet and passes it to the HPCC++ module for processing. A telemetry response packet can be an ACK containing inband telemetry, or a dedicated telemetry response prob packet.

2. Receiver side flow

On receiving a packet containing inband telemetry, the RX pipeline extracts the network status, and the flow parameters from the packet and passes it to the TX pipeline. The packet can be a data packet containing inband telemetry, or a dedicated telemetry request prob packet. The Tx pipeline may process and edit the telemetry data, and then sends back to the sender the data using either an ACK packet of the flow or a dedicated telemetry response packet.

6.3.2. Receiver-based HPCC

Note that the window/rate calculation can be implemented at either the data sender or the data receiver. If the ACK packets already exist for reliability purpose, the inband telemetry information can be echoed back to the sender via ACK self-clocking. Not all ACK packets need to carry the inband telemetry information. To reduce the Packet Per Second (PPS) overhead, the receiver may examine the inband telemetry information and adopt the technique of delayed ACKs that only sends out an ACK for a few of received packets. In order to reduce PPS even further, one may implement the algorithm at the receiver and feedback the calculated window in the ACK packet once every RTT.

The receiver-based algorithm, Rx-HPCC, is based on int.L, which is the inband telemetry information in the packet header. The receiver performs the same functions except using int.L instead of ack.L. The new function NewINT(int.L) is to replace NewACK(int.L)

```
28:   Procedure NewINT(int.L)
29:   if now > (lastUpdateTime + T) then
30:      W = ComputeWind(MeasureInflight(int), True);
31:      send_ack(W)
32:      lastUpdateTime = now;
33:   else
34:      W = ComputeWind(MeasureInflight(int), False);
```

Here, since the receiver does not know the starting sequence number of a burst, it simply records the lastUpdateTime. If time T has passed since lastUpdateTime, the algorithm would recalcuate Wc as in Line 30 and send out the ACK packet which would include W information. Otherwise, it would just update W information locally. This would reduce the amount of traffic that needs to be feedback to the data sender.

Note that the receiver can also measure the number of outstanding flows, N, if the last hop is the congestion point and use this information to dynamically adjust W_ai to achieve better fairness. The improvement would allow flows to quickly converge to fairness without causing large swings under heavy load.

7. Reference Implementation

HPCC++ can be adopted as the CC algorithm by a wide range of transport protocols such as TCP and UDP, as well as others that may run on top of them, such as iWARP, RoCE etc. It requires to have the window limit and congestion feedback through ACK self-clocking, which naturally conforms to the paradigm of TCP design. With that, HPCC++ introduces a scheme to measure the total inflight bytes for more precise congestion control. To run in UDP, some modifications need to be done to enforce the window limit and collect congestion feedback via probing packets, which is incremental.

7.1. Implementation on RDMA RoCEv2

We describe reference implementation on RDMA RoCEv2. This is an implementation for ``Sender-based HPCC++'' (see section 6.3.1.) using dedicated probe packets to collect the telemetry. HPCC++ module in the sender triggers the sending of ``telemetry request packet'' for a given flow. The NIC then sends the probe packet. The packet will have the same IP and UDP headers as the data packets of the given flow. Such packet is expected to be sent every RTT, see section 6 for more details. On receiving of telemetry request packet, the NIC extracts the telemetry from all the links along the path from the sender. HPCC++ module chooses the link with the highest inflight bytes and sends its telemetry (queue length, timestamp and tx bytes) back to the receiver on top of dedicated ``telemetry response

packet''. On receiving of telemetry response packet, the NIC extracts the telemetry and pass it to the HPCC++ module which using this info to implement the rate update scheme.

7.2. Implementation on TCP

Taking the benefit of precise congestion control for TCP is a natural next step. Since TCP segmentation at TX side (e.g., TSO) and coalescing at RX side (e.g., GRO) happen at the NIC HW or low-layer of TCP/IP stack, carrying per-pkt inband telemetry info between the TCP congestion control engine and network fabric has to work with the TSO and GRO. Instead, one way to adopt HPCC++ for TCP is using the special probe and notification packets to retrieve inband telemetry information. The sender generates a probe packet when it is actively sending data. The probe packet has the same 5-tuples (source and destination addresses, source and destination ports and protocol number) with the data packets and the inband telemetry header. The switches along the path identify the probe packet by its inband telemetry header and insert the inband telemetry. Once received the probe packet with inband telemetry, the receiver replies with a response packet piggybacking the inband telemetry to the sender. Note, both probe and response packets use a special DSCP number so that it can bypass the TSO and GRO in each side.

8. IANA Considerations

This document makes no request of IANA.

9. Discussion

9.1. Internet Deployment

Although the discussion above mainly focuses on the data center environment, HPCC++ can be adopted at Internet at large. There are several security considerations one should be aware of.

There may rise privacy concern when the telemetry information is conveyed across Autonomous Systems (ASes) and back to end-users. The link load information captured in telemetry can potentially reveal the provider's network capacity, route utilization, scheduling policy, etc. Those usually are considered to be sensitive data of the network providers. Hence, certain action may take to anonymize the telemetry data and only convey the relative ratio in rate adaptation across ASes without revealing the actual network load.

Another consideration is the security of receiving telemetry information. The rate adaptation mechanism in HPCC++ relies on feedback from the network. As such, it is vulnerable to attacks

where feedback messages are hijacked, replaced, or intentionally injected with misleading information resulting in denial of service, similar to those that can affect TCP. It is therefore RECOMMENDED that the notification feedback message is at least integrity checked. In addition, [I-D.ietf-avtcore-cc-feedback-message] discusses the potential risk of a receiver providing misleading congestion feedback information and the mechanisms for mitigating such risks.

9.2. Switch-assisted congestion control

HPCC++ falls in the general category of switch-assisted congestion control. However, HPCC++ includes a few unique design choices that are different from other switch-assisted approaches.

- First, HPCC++ implements a primal-mode algorithm that requires only the ``write-to-packet'' operation from switches, which has already been supported by telemetry protocols like INT [P4-INT] or IOAM [I-D.ietf-ippm-ioam-data]. Please note that this is very different from dual-mode algorithms such as XCP [Katabi-SIGCOMM2002] and RCP [Dukkipati-RCP], where switches take an actively role in determining flows' rates.

- Second, HPCC++ achieves a fast utilization convergence by decoupling it from fairness convergence, which is inspired by XCP.

- Third, HPCC++ enables the switch-guided multiplicative increase (MI) by defining the ``inflight byte'' to quantify the link load. The inflight byte tells both the underload and overload of the link precisely and thus it allows the flow to increase/decrease the rate multiplicatively and safely. By contrast, traditional approaches of using the queue length or RTT as the feedback cannot guide the rate increase and instead have to rely on additive increase (AI) with heuristics. As the link speed contines to grow, this becomes increasingly slow in reclaiming the unused bandwidth. Besides, queue-based feedback mechanisms subject to latency inflation.

- Last, HPCC++ uses TX rate instead of RX rate used by XCP and RCP. As detailed in [SIGCOMM-HPCC], we view the TX rate is more precise because RX rate and queue length are overlapped and thus it causes oscillation.

9.3. Work with QoS queuing

Under the use of QoS (Quality of service) priority queuing in switches, the length of flow's own queue cannot tell the actual queuing time and the exact extent of congestion. Although general approaches for running congestion control with QoS queuing are out of

the scope of this document, we provide a few hints for HPCC++ running friendly with QoS queuing. In this case, HPCC++ can leverage the packet sojourn time (the egress timestamp minus the ingress timestamp) instead of the queue length to quantify the packet's actual queuing delay. In addition, the operators typically use the Deficit Weighted Round Robin (DWRR) instead of the strict priority (SP) as their QoS scheduling to prevent traffic starvation. DWRR provides a minimum bandwdith guarantee for each queue so that HPCC++ can leverage it for precise rate update to avoid congestion.

9.4. Path migration

HPCC++ allows switches and end-hosts to share precise information of network utilization, which suggests a framework for path selection and rate control at end-hosts. The framework HPCC++ enabled is to leverage each switch to report its link load information via inband telemetry. The end-host fetches inband telemetry along the traffic routes and makes a timely and accurate decision on path selection and traffic admission.

10. Acknowledgments

The authors would like to thank RTGWG members for their valuable review comments and helpful input to this specification.

11. Contributors

The following individuals have contributed to the implementation and evaluation of the proposed scheme, and therefore have helped to validate and substantially improve this specification: Pedro Y. Segura, Roberto P. Cebrian, Robert Southworth and Malek Musleh.

12. References

12.1. Normative References

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

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

12.2. Informative References

[Dukkipati-RCP]
Dukkipati, N., "Rate Control Protocol (RCP): Congestion control to make flows complete quickly.", Stanford University , 2008.

[I-D.ietf-avtcore-cc-feedback-message]
Sarker, Z., Perkins, C., Singh, V., and M. A. Ramalho, "RTP Control Protocol (RTCP) Feedback for Congestion Control", draft-ietf-avtcore-cc-feedback-message-09 (work in progress), November 2020.

[I-D.ietf-ippm-ioam-data]
"Data Fields for In-situ OAM", March 2020, <https://tools.ietf.org/html/draft-ietf-ippm-ioam-data-09>.

[I-D.ietf-kumar-ippm-ifa]
"Inband Flow Analyzer", February 2019, <https://tools.ietf.org/html/draft-kumar-ippm-ifa-01>.

[Katabi-SIGCOMM2002]
Katabi, D., Handley, M., and C. Rohrs, "Congestion Control for High Bandwidth-Delay Product Networks", ACM SIGCOMM Pittsburgh, Pennsylvania, USA, October 2002.

[P4-INT] "In-band Network Telemetry (INT) Dataplane Specification, v2.0", February 2020, <https://github.com/p4lang/p4-applications/blob/master/docs/INT_v2_0.pdf>.

[SIGCOMM-HPCC]
Li, Y., Miao, R., Liu, H., Zhuang, Y., Fei Feng, F., Tang, L., Cao, Z., Zhang, M., Kelly, F., Alizadeh, M., and M. Yu, "HPCC: High Precision Congestion Control", ACM SIGCOMM Beijing, China, August 2019.

[Zhu-SIGCOMM2015]
Zhu, Y., Eran, H., Firestone, D., Guo, C., Lipshteyn, M., Liron, Y., Padhye, J., Raindel, S., Yahia, M., and M. Zhang, "Congestion Control for Large-Scale RDMA Deployments", ACM SIGCOMM London, United Kingdom, August 2015.

Authors' Addresses

Rui Miao
Alibaba Group
525 Almanor Ave, 4th Floor
Sunnyvale, CA 94085
USA

Email: miao.rui@alibaba-inc.com

Hongqiang H. Liu
Alibaba Group
108th Ave NE, Suite 800
Bellevue, WA 98004
USA

Email: hongqiang.liu@alibaba-inc.com

Rong Pan
Intel, Corp.
2200 Mission College Blvd.
Santa Clara, CA 95054
USA

Email: rong.pan@intel.com

Jeongkeun Lee
Intel, Corp.
4750 Patrick Henry Dr.
Santa Clara, CA 95054
USA

Email: jk.lee@intel.com

Changhoon Kim
Intel Corporation
4750 Patrick Henry Dr.
Santa Clara, CA 95054
USA

Email: chang.kim@intel.com

Barak Gafni
Mellanox Technologies, Inc.
350 Oakmead Parkway, Suite 100
Sunnyvale, CA 94085
USA

Email: gbarak@mellanox.com

Yuval Shpigelman
Mellanox Technologies, Inc.
Haim Hazaz 3A
Netanya 4247417
Israel

Email: yuvals@nvidia.com

Jeff Tantsura
Microsoft Corporation
One Microsoft Way
Redmond, Washington 98052-6399
USA

Email: jefftantsura@microsoft.com

Network Working Group
Internet-Draft
Intended status: Informational
Expires: 19 March 2024

S. Peng
Z. Li
Huawei Technologies
G. Mishra
Verizon Inc.

16 September 2023

APN Scope and Gap Analysis
draft-peng-apn-scope-gap-analysis-07

Abstract

The APN work in IETF is focused on developing a framework and set of mechanisms to derive, convey and use an attribute allowing the implementation of fine-grain user group-level and application group-level requirements in the network layer. APN aims to apply various policies in different nodes along a network path onto a traffic flow altogether, for example, at the headend to steer into corresponding path, at the midpoint to collect corresponding performance measurement data, and at the service function to execute particular policies. Currently there is still no way to efficiently realize this composite network service provisioning along the path. This document further clarifies the scope of the APN work and describes the solution gap analysis.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

Application-aware Networking (APN) is introduced in [I-D.li-apn-framework] and [I-D.li-apn-problem-statement-usecases]. APN conveys an attribute along with data packets into network and makes the network aware about data flow requirements at different granularity levels.

Such an attribute is acquired, constructed in a structured value, and then encapsulated in the packet. Such structured value is treated as an opaque object in the network to which the network operator applies policies in various nodes/service functions along the path and provides corresponding services.

This structured attribute can be encapsulated in various data planes adopted within a Network Operator controlled limited domain, e.g. MPLS, VXLAN, SR/SRv6 and other tunnel technologies, which waits to be further specified.

With APN, it becomes possible to apply various policies in different nodes along a network path onto a traffic flow altogether in a more efficient way, e.g., at the headend to steer into corresponding path, at the midpoint to collect corresponding performance measurement data, and at the service function to execute particular policies. Currently there is still no way to realize this composite network service provisioning along the path very efficiently. It may be possible to stack those various policies in a list of TLVs at the headend. However, this approach would introduce great complexities and impose big challenges on the hardware processing and forwarding.

The example use-case presented in this draft further expands on the rationale for such an attribute and how it can be derived and used in that specific context.

This document further clarifies the scope of the APN work and describes the solution gap analysis.

2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 RFC 2119 [RFC2119] RFC 8174 [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Terminologies

APN: Application-aware Networking

CPE: Customer Premises Equipment

DPI: Deep Packet Inspection

OS: Operating System

4. APN Framework and Scope

The APN framework is introduced in [I-D.li-apn-framework], as shown in the Figure 1.

```
                            +------------------+
                            |   APN-Customer   |
                            +------------------+
                                     |
                                     | NBI
                                     |
                            +------------------+
                     -------|  APN-Controller  |--------
                   /        +------------------+        \
Client            /        /         |          \        \           Server
+-----+          /        /          | SBI       \        \          +-----+
|App x|-\       |        |           |            |        |      /->|App x|
+-----+ |   +-----+ +-----+   +---------+   +--------+ +-----+  |  +-----+
         \->|APN  | |APN  |-A-|APN      |-A-|APN     | |APN  |->/
User side   |-    |-|-    |-B-|-        |-B-|-       |-|-    |
         /->|Edge | |Head |-C-|Midpoint |-C-|Endpoint| |Edge |->\
+-----+ |   +-----+ +-----+   +---------+   +--------+ +-----+  |   +-----+
|App y|-/      |------------------APN Domain--------------|      \->|App y|
+-----+         Network Operator Controlled Limited Domain          +-----+
```

Figure 1. APN Framework and Scope

APN is only applied to an edge-to-edge tunnel encapsulation within a limited trusted domain. It means that the source and destination addresses of the packet are the endpoints of the tunnel (i.e. the domain edges), and nothing about the payload source and destination can be deduced, which substantially reduces the privacy concerns. Typically, an APN domain is defined as a Network Operator controlled limited domain (see Figure 1), in which MPLS, VXLAN, SR/SRv6 and other tunnel technologies are adopted to provide network services.

With APN, the attribute is acquired based on the existing information in the packet header (i.e. source and destination addresses, incoming L2 (or) MPLS encapsulation, incoming physical/virtual port information, the other fields of the 5-tuple if they are not encrypted) at the edge devices of the APN domain, added to the data packets along with the tunnel encapsulation, and delivered to the network, wherein, according to this attribute, corresponding network services are provisioned. When the packets leave the APN domain, the attribute is removed together with the tunnel encapsulation header.

5. Example Use Case and Existing Issues

To be more specific and more concrete, here we use SD-WAN as an example use case to further expand on the rationale for such attribute and how it can be derived and used in that specific context.

In the case of SD-WAN, an enterprise obtains WAN services from an SD-WAN provider so that its employees have access to the applications in the Cloud, and then the SD-WAN provider may buy WAN lines from a Network Operator. The enterprise may know what applications will use the SD-WAN services, but it will only provide the 5 tuples (i.e. source IP address, source port, destination IP address, destination port, transport protocol) of those applications to the SD-WAN provider. So, the SD-WAN provider does not know what applications it is serving, and will only provide 5 tuples to the Network Operator and the service performance requirements for steering their customer's traffic. In this way, the Network Operator does not know anything else about the traffic except the 5 tuples and requirements. Nowadays, SD-WAN is usually using 5-tuple to steer the traffic into corresponding WAN lines across the Network Operator's network [SD-WAN].

However, there are two main issues in the current SD-WAN deployments.

1) It is complicated to resolve the 5 tuples. Even worse, as the traffic is encrypted, it becomes impossible to obtain any transport layer information. Moreover, in the IPv6 data plane, with the extension headers being added before the upper layer, in some implementations it becomes very difficult and even impossible to obtain transport layer information because that information is located deep in the packet. So, there is no 5 tuples anymore, and maybe only 2 tuples are available.

2) Currently there is still no way to apply various policies in different nodes along the network path onto a traffic flow altogether, that is, at the headend to steer into corresponding path, at the midpoint to collect corresponding performance measurement data, and at the service function to execute particular policies. It may be possible to stack those various policies in a list of TLVs at the headend. However, this approach would introduce great complexities and impose big challenges on the hardware processing and forwarding.

6. Basic Solution and Benefits

With APN, at the edge node, i.e. CPE, of the SD-WAN (see Figure 2), the 5-tuple, plus information related to user or application group-level requirements is constructed into a structured value, called APN attribute. This attribute is only meaningful for the network operators to apply various policies in different nodes/service functions, which can be enforced from the Controllers.

```
                          +-----------------+
                +---------|SD-WAN Controller|---------+
                |         +--------|--------+         |
                |                  |                  |
                |          +-------|-------+          |
                |          |SDN  Controller|          |
                |          +-------|-------+          |
+-----+         |                  |                  |       +-----+
|App x|-\       |                  |                  |     /-|App x|
+-----+ |  +--|--+   +-----------|-----------+   +--|--+ | +-----+
         \-|     |   |   Application-aware   |   |     |-/
           |CPE 1|---|        Network        |---|CPE 2|
         /-|     |   |  Service Provisioning |   |     |-\
+-----+ |  +-----+   +-----------------------+   +-----+ | +-----+
|App y|-/       |                                  |     \-|App y|
+-----+         |<--- Network Operator Controlled --->|      +-----+
                            Limited Domain
```

Figure 2. SD-WAN using the APN Framework

With such an attribute in the network, we can easily solve the two issues above-mentioned. For example, when the packet is sent from the CPE1 and the attribute is added along with the tunnel encapsulation, then it is not necessary to resolve the 5-tuple and perform the deep inspection in every node along the path. This attribute is encapsulated in the network layer and can be easily read by the routers and service functions. If the tunnel is based on the IPv6 data plane, for example, such an attribute can be encapsulated in an option of IPv6 hop-by-hop options header.

Since this attribute is taken as an object to the network, the network operators will simply place the policies in the nodes/service functions where this indicated traffic will go through, and the corresponding node/service function will just apply policies for this object. This can be easily done by utilizing this attribute, which is not possible with any current existing mechanism.

Such attribute will also bring other benefits, for example,

* Improve the forwarding performance since it will only use 1 field in the IP layer instead of resolving 5 tuples, which will also improve the scalability.

* Very flexible policy enforcement in various nodes and service functions along the network path.

Furthermore, with such attribute, more new services could be enabled, for example,

* Even more fine-granularity performance measurement could be achieved and the granularity to be monitored and visualized can be controllable, which is able to relieve the processing pressure on the controller when it is facing the massive monitoring data.

* The policy execution on the service function can be based only on this value and not based on 5-tuple, which can eliminate the need of deep packet inspection.

* The underlay performance guarantee could be achieved for SD-WAN overlay services, such as explicit traffic engineering path satisfying SLA and selective visualized accurate performance measurement.

7. Solution Gap Analysis

There are already some solutions specified in IETF, which use identifier to perform traffic steering and service provisioning. However, the existing solutions are specific to a particular scenario or data plane. None of them is the same as APN and able to achieve the same effects.

7.1. IPv6/MPLS Flow Label

[RFC6437] specifies the IPv6 flow label which enables the IPv6 flow classification. However, the IPv6 flow label is mainly used for Equal Cost Multipath Routing (ECMP) and Link Aggregation [RFC6438].

Similarly, [RFC6391] describes a method of adding an additional Label Stack Entry (LSE) at the bottom of the stack in order to facilitate the load balancing of the flows within a pseudowire (PW) over the available ECMPs. A similar design for general MPLS use has also been proposed in [RFC6790] using the concept of Entropy Label.

7.2. Subscriber Identifier and Performance Policy Identifier in NSH

Subscriber Identifier and Performance Policy Identifier are specified in [RFC8979]. These identifiers are carried only in the Network Service Header (NSH) [RFC8300] Context Header, as shown in Figure 3, while the APN attribute can be carried in various data plane encapsulations.

```
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Metadata Class       |      Type     |U|    Length   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
~                       Subscriber Identifier                   ~
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+


0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Metadata Class       |      Type     |U|    Length   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
~                     Performance Policy Identifier             ~
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3. Subscriber Identifier and Performance Policy Identifier

In this draft [RFC8979], the Subscriber Identifier carries an opaque local identifier that is assigned to a subscriber by a network operator, and the Performance Policy Identifier represents an opaque value pointing to specific performance policy to be enforced. In this way, in order to apply various policies in different nodes along the network path onto a traffic flow altogether, e.g., at the headend to steer into corresponding path, at the midpoint to collect corresponding performance measurement data, and at the service function to execute particular policies, those various policies would have to be stacked in a list of TLVs at the headend, introducing great complexities and big challenges on the hardware processing and forwarding.

The APN attribute is treated as an opaque object in the network, to which the network operator applies policies in various nodes/service functions along the path and provide corresponding services.

7.3. IOAM Flow ID

A 32-bit Flow ID is specified in [I-D.ietf-ippm-ioam-direct-export], which is used to correlate the exported data of the same flow from multiple nodes and from multiple packets, while the APN attribute can serve more various purposes.

7.4. Binding SID

The Binding SID (BSID) [RFC8402] is bound to an SR Policy, instantiation of which may involve a list of SIDs. Any packets received with an active segment equal to BSID are steered onto the bound SR Policy. A BSID may be either a local or a global SID. While the APN attribute is not bound to SR only, and it can be carried in various data plane encapsulations.

7.5. FlowSpec Label

The flow specification (FlowSpec) [RFC5575] is actually an n-tuple consisting of several matching criteria that can be applied to IP traffic, which include elements such as source and destination address prefixes, IP protocol, and transport protocol port numbers. In BGP VPN/MPLS networks, BGP FlowSpec can be extended to identify and change (push/swap/pop) the label(s) for traffic that matches a particular FlowSpec rule in [I-D.ietf-idr-flowspec-mpls-match] and [I-D.ietf-idr-bgp-flowspec-label]. In [I-D.liang-idr-bgp-flowspec-route], BGP is used to distribute the FlowSpec rule bound with label(s). While the APN attribute is not bound to MPLS only, and it can be carried in various data plane encapsulations.

7.6. Group Policy ID

The capabilities of the VXLAN-GPE protocol can be extended by defining next protocol "shim" headers that are used to implement new data plane functions. For example, Group Policy ID is carried in the Group-Based Policy (GBP) Shim header [I-D.lemon-vxlan-lisp-gpe-gbp]. GENEVE has similar ability as VXLAN-GPE to carry metadata.

7.7. Detnet Flow Identification

Identification and Specification of DetNet Flows is specified in [RFC9016]. DetNet MPLS flows can be identified and specified by the SLabel and the FLabelStack. The IP 6-tuple is used for DetNet IP flow identification, which consists of SourceIpAddress, DestinationIpAddress, Dscp, Protocol, SourcePort, and DestinationPort. IPv6FlowLabel and IPSecSpi are additional attributes that can be used for DetNet flow identification in

addition to the 6-tuple. Therefore, the Detnet IP Flow ID is logical and there is no such Flow ID carried for Detnet, but only the 6-tuple is directly used to identify the Detnet flows.

Only one exceptional case, in [I-D.ietf-spring-sr-redundancy-protection], the 32-bit flow identification (FID) identifies one specific Detnet flow of redundancy protection. This FID is usually allocated from centralized controller to the SR ingress node or redundancy node in SR network.

7.8. Network Slicing Resource ID

In [I-D.ietf-6man-enhanced-vpn-vtn-id], VTN Resource ID is a 4-octet identifier which uniquely identifies the set of network resources allocated to a VTN. For network slicing, the ID is used to indicate the network resources to be allocated to the network slices and it is not bound to any traffic flow.

APN is for traffic steering, while network slicing is about resource partition [I-D.ietf-teas-rfc3272bis].

7.9. Service Path ID in NSH

In [RFC8300], the Service Path Identifier (SPI) in the NSH uniquely identifies a Service Function Path (SFP). Participating nodes MUST use this identifier for SFP selection. The initial Classifier MUST set the appropriate SPI for a given classification result. For SFC, the ID is used to indicate a SF path and it is not bound to any traffic flow.

7.10. DSCP

DSCP [RFC4594] is mainly used for per hop QoS handling inside the network. It has 6 bits to set.

7.11. Summary

The comparison of the identifiers for the typical network services (incl. iOAM, Detnet, Network Slicing (NS), and Service Function Chaining (SFC)) is shown in the following Table from different aspects (incl. ID, Identification Object, Source (for generating the ID), Configuration (Conf.) node, and Size).

	ID	Identification Object	Source	Conf. node	Size
APN	APN ID	The flow that needs fine-granular services	5-tuple Layer 2	Controller	32bits 128b
iOAM	Flow ID	The flow that needs performance monitoring	-	Controller Ingress	32bits
Detnet	Flow ID (6-tuple)	The flow that needs Detnet services	-	Controller	-
Detnet	Flow ID	The redundant protection flow	-	Detnet Controller	32bits
NS	Resource ID	The network resources that are allocated to network slices	-	Controller	32bits
NSH	SPI	The SF Path	-	Controller	24bits
SFC	Performance Policy ID	The performance policy	-	Controller	-
DSCP		Per-hop Behavior	-		6bits

Table 1. Comparison of the Identifiers

As driven by ever-emerging new 5G services, fine-granularity service provisioning becomes urgent. The existing solutions are either specific to a particular scenario or data plane. While APN aims to define a generalized attribute used for fine-granularity service provisioning, and can be carried in various data plane encapsulations.

8. IANA Considerations

There are no IANA considerations in this document.

9. Acknowledgements

The authors would like to acknowledge Martin Vigoureux, Alvaro Retana, Barry Leiba, Stefano Previdi, Adrian Farrel, and Daniel King for their valuable review and comments.

10. Informative References

[I-D.brockners-ippm-ioam-vxlan-gpe]
Brockners, F., Bhandari, S., Govindan, V. P., Pignataro, C., Gredler, H., Leddy, J., Youell, S., Mizrahi, T., Kfir, A., Gafni, B., Lapukhov, P., and M. Spiegel, "VXLAN-GPE Encapsulation for In-situ OAM Data", Work in Progress, Internet-Draft, draft-brockners-ippm-ioam-vxlan-gpe-03, 4 November 2019, <https://datatracker.ietf.org/doc/html/draft-brockners-ippm-ioam-vxlan-gpe-03>.

[I-D.ietf-6man-enhanced-vpn-vtn-id]
Dong, J., Li, Z., Xie, C., Ma, C., and G. S. Mishra, "Carrying Virtual Transport Network (VTN) Information in IPv6 Extension Header", Work in Progress, Internet-Draft, draft-ietf-6man-enhanced-vpn-vtn-id-05, 6 July 2023, <https://datatracker.ietf.org/doc/html/draft-ietf-6man-enhanced-vpn-vtn-id-05>.

[I-D.ietf-idr-bgp-flowspec-label]
liangqiandeng, Hares, S., You, J., Raszuk, R., and D. Ma, "Carrying Label Information for BGP FlowSpec", Work in Progress, Internet-Draft, draft-ietf-idr-bgp-flowspec-label-02, 20 October 2022, <https://datatracker.ietf.org/doc/html/draft-ietf-idr-bgp-flowspec-label-02>.

[I-D.ietf-idr-flowspec-mpls-match]
Yong, L., Hares, S., liangqiandeng, and J. You, "BGP Flow Specification Filter for MPLS Label", Work in Progress, Internet-Draft, draft-ietf-idr-flowspec-mpls-match-02, 20 October 2022, <https://datatracker.ietf.org/doc/html/draft-ietf-idr-flowspec-mpls-match-02>.

[I-D.ietf-ippm-ioam-direct-export]
Song, H., Gafni, B., Brockners, F., Bhandari, S., and T. Mizrahi, "In Situ Operations, Administration, and Maintenance (IOAM) Direct Exporting", Work in Progress, Internet-Draft, draft-ietf-ippm-ioam-direct-export-11, 23 September 2022, <https://datatracker.ietf.org/doc/html/draft-ietf-ippm-ioam-direct-export-11>.

[I-D.ietf-sfc-serviceid-header]
Sarikaya, B., Von Hugo, D., and M. Boucadair, "Subscriber and Performance Policy Identifier Context Headers in the Network Service Header (NSH)", Work in Progress, Internet-Draft, draft-ietf-sfc-serviceid-header-14, 11 December 2020, <https://datatracker.ietf.org/doc/html/draft-ietf-sfc-serviceid-header-14>.

[I-D.ietf-spring-sr-redundancy-protection]
Geng, X., Chen, M., Yang, F., Camarillo, P., and G. S. Mishra, "SRv6 for Redundancy Protection", Work in Progress, Internet-Draft, draft-ietf-spring-sr-redundancy-protection-02, 23 September 2022, <https://datatracker.ietf.org/doc/html/draft-ietf-spring-sr-redundancy-protection-02>.

[I-D.ietf-teas-rfc3272bis]
Farrel, A., "Overview and Principles of Internet Traffic Engineering", Work in Progress, Internet-Draft, draft-ietf-teas-rfc3272bis-27, 12 August 2023, <https://datatracker.ietf.org/doc/html/draft-ietf-teas-rfc3272bis-27>.

[I-D.lemon-vxlan-lisp-gpe-gbp]
Lemon, J., Maino, F., Smith, M., and A. Isaac, "Group Policy Encoding with VXLAN-GPE and LISP-GPE", Work in Progress, Internet-Draft, draft-lemon-vxlan-lisp-gpe-gbp-02, 30 April 2019, <https://datatracker.ietf.org/doc/html/draft-lemon-vxlan-lisp-gpe-gbp-02>.

[I-D.li-apn-framework]
Li, Z., Peng, S., Voyer, D., Li, C., Liu, P., Cao, C., and G. S. Mishra, "Application-aware Networking (APN) Framework", Work in Progress, Internet-Draft, draft-li-apn-framework-07, 3 April 2023, <https://datatracker.ietf.org/doc/html/draft-li-apn-framework-07>.

[I-D.li-apn-problem-statement-usecases]
Li, Z., Peng, S., Voyer, D., Xie, C., Liu, P., Qin, Z., and G. S. Mishra, "Problem Statement and Use Cases of Application-aware Networking (APN)", Work in Progress, Internet-Draft, draft-li-apn-problem-statement-usecases-08, 3 April 2023, <https://datatracker.ietf.org/doc/html/draft-li-apn-problem-statement-usecases-08>.

[I-D.liang-idr-bgp-flowspec-route]
Liang, Q. and J. You, "BGP FlowSpec based Multi-dimensional Route Distribution", Work in Progress, Internet-Draft, draft-liang-idr-bgp-flowspec-route-00, 20 October 2014, <https://datatracker.ietf.org/doc/html/draft-liang-idr-bgp-flowspec-route-00>.

[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>.

[RFC4594] Babiarz, J., Chan, K., and F. Baker, "Configuration Guidelines for DiffServ Service Classes", RFC 4594, DOI 10.17487/RFC4594, August 2006, <https://www.rfc-editor.org/info/rfc4594>.

[RFC5575] Marques, P., Sheth, N., Raszuk, R., Greene, B., Mauch, J., and D. McPherson, "Dissemination of Flow Specification Rules", RFC 5575, DOI 10.17487/RFC5575, August 2009, <https://www.rfc-editor.org/info/rfc5575>.

[RFC6391] Bryant, S., Ed., Filsfils, C., Drafz, U., Kompella, V., Regan, J., and S. Amante, "Flow-Aware Transport of Pseudowires over an MPLS Packet Switched Network", RFC 6391, DOI 10.17487/RFC6391, November 2011, <https://www.rfc-editor.org/info/rfc6391>.

[RFC6437] Amante, S., Carpenter, B., Jiang, S., and J. Rajahalme, "IPv6 Flow Label Specification", RFC 6437, DOI 10.17487/RFC6437, November 2011, <https://www.rfc-editor.org/info/rfc6437>.

[RFC6438] Carpenter, B. and S. Amante, "Using the IPv6 Flow Label for Equal Cost Multipath Routing and Link Aggregation in Tunnels", RFC 6438, DOI 10.17487/RFC6438, November 2011, <https://www.rfc-editor.org/info/rfc6438>.

[RFC6790] Kompella, K., Drake, J., Amante, S., Henderickx, W., and L. Yong, "The Use of Entropy Labels in MPLS Forwarding", RFC 6790, DOI 10.17487/RFC6790, November 2012, <https://www.rfc-editor.org/info/rfc6790>.

[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>.

[RFC8300] Quinn, P., Ed., Elzur, U., Ed., and C. Pignataro, Ed., "Network Service Header (NSH)", RFC 8300, DOI 10.17487/RFC8300, January 2018, <https://www.rfc-editor.org/info/rfc8300>.

[RFC8402] Filsfils, C., Ed., Previdi, S., Ed., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", RFC 8402, DOI 10.17487/RFC8402, July 2018, <https://www.rfc-editor.org/info/rfc8402>.

[RFC8979] Sarikaya, B., von Hugo, D., and M. Boucadair, "Subscriber and Performance Policy Identifier Context Headers in the Network Service Header (NSH)", RFC 8979, DOI 10.17487/RFC8979, February 2021, <https://www.rfc-editor.org/info/rfc8979>.

[RFC9016] Varga, B., Farkas, J., Cummings, R., Jiang, Y., and D. Fedyk, "Flow and Service Information Model for Deterministic Networking (DetNet)", RFC 9016, DOI 10.17487/RFC9016, March 2021, <https://www.rfc-editor.org/info/rfc9016>.

[SD-WAN] MEF 70.1 Draft (R1), available at https://www.mef.net/wp-content/uploads/2020/08/MEF-70-1-Draft-R1.pdf/, "SD-WAN Service Attributes and Service Framework", August 2020.

Authors' Addresses

Shuping Peng
Huawei Technologies
Beijing
China
Email: pengshuping@huawei.com

Zhenbin Li
Huawei Technologies
Beijing
China
Email: lizhenbin@huawei.com

Gyan Mishra
Verizon Inc.
United States of America
Email: gyan.s.mishra@verizon.com

Network Working Group
Internet-Draft
Intended status: Informational
Expires: 12 August 2024

D. Trossen
Huawei Technologies
D. Guzman
Technical University Munich
M. McBride
FutureWei
X. Fan
IoTeX

9 February 2024

Impact of DLTs on Provider Networks
draft-trossen-rtgwg-impact-of-dlts-03

Abstract

This document discusses the impact of distributed ledger technologies being realized over IP-based provider networks. The focus here lies on the impact that the DLT communication patterns have on efficiency of resource usage in the underlying networks. We provide initial insights into experimental results to quantify this impact in terms of inefficient and wasted communication, aligned along challenges that the DLT realization over IP networks faces.

This document intends to outline this impact but also opportunities for network innovations to improve on the identified impact as well as the overall service quality. While this document does not promote specific solutions that capture those opportunities, it invites the wider community working on DLT and network solutions alike to contribute to the insights in this document to aid future research and development into possible solution concepts and technologies.

The findings presented here have first been reported within the similarly titled whitepaper released by the Industry IoT Consortium (IIC) [IIC_whitepaper].

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

The current routing system was initially designed for a single purpose, namely reachability between end nodes. This capability is utilized in many higher layer technologies in the form of overlays. Distributed Ledger Technologies (DLT) are one such form of overlay with the aim to facilitate communication patterns that allow a distributed consensus among distributed, and generally unknown, participants in the DLT overlay.

The realization of a DLT overlay follows that of other well-known examples for distributed computing tasks, such as Torrents, distributed file storage, among others. That is, DLTs form their own overlay through contributing 'peers' that partake in the DLT. For this, reachability information (in the form of IP addresses) of other DLT peers is centrally maintained (in so-called 'bootstrap nodes') to establish peer-specific pools of peers, within which each peer in turn communicates for the specific purpose of the DLT. DLTs secure the transactions using transport-level methods. As an overlay, DLTs are little concerned with the underlying network(s) itself, simply utilizing the provided IP reachability service for their purpose.

Continuing on the insights first reported in [IIC_whitepaper], this document sheds light onto the realization of specific DLT overlay mechanisms from the perspective of the resulting impact on the utilized provider networks in the form of the actual communication taking place.

For this, we outline the communication patterns upon which certain forms of DLTs rely (Section 4.2) in order to implement the key DLT concepts (Section 3). Based on our insights of those communication patterns, we then identify a number of key challenges (Section 5) through initial experimental results (Section 6) within an example DLT platform (here, Ethereum [REF]).

Here, we explicitly recognize that those insights are highly dependent on the specific DLT mechanisms under investigation and are therefore not generally transferable to other DLT platforms and their realization. For instance, DLT platforms relying on proof-of-work for transaction verification tend to differ in their communication from those relying on proof-of-stake. However, this document does attempt to develop a wider methodology over time that may allow for quantifying the impact on underlying networks across those different types of DLTs.

While the quantification of DLT impact serves as an interesting benchmark into the possible costs for operating DLTs, the identified challenges give also rise to possible opportunities for network-level

innovations to improve on the situation observed in our experiments, thereby reducing the identified impact on provider network. Section 7 outlines a possible realization of those opportunities through a constraint-based selection of communication relations, utilizing semantic information beyond IP reachability.

With this in mind, we position an improved DLT performance as a possible applicability for semantic routing, introduced in more detail in [I-D.farrel-irtf-introduction-to-semantic-routing], while also soliciting other possible realizations of an improved DLT performance through network-level innovations. Moreover, we draw connections with ongoing IETF/IRTF efforts (Section 8), where our insights may provide useful input.

Note: This document does neither discuss the particular rationale for selecting DLTs in order to realize the intended application purpose nor the specific DLT mechanisms eventually used. It therefore does not pass comment on the advisability or practicality of using DLTs and their solutions, nor does it define any specific technical solutions for reducing the observed provider impact.

2. Terminology

The following terminology is used throughout the remainder of this draft:

Smart contract : distributed state machine over which transactions will be executed and logged.

Transaction : cryptographically signed (set of) instruction(s) against a smart contract.

Ledger : information on transactions

Block : set of verified ledger information

Blockchain : concatenated blocks with longest chain of blocks representing the current consensus of ledger information.

Peer : participant in the DLT, with a possible narrower role of client or miner.

Client : a DLT peer issuing transactions towards a set of miners.

Miner : a DLT peer receiving transactions from miners and

performing suitable block operations and exchanges to maintain DLT information.

3. Main DLT Concepts

There has been ample work, such as [DLT_intro] [DLT_intro2], among others, including in other SDOs such as the IEEE but also within the IRTF/IETF [DINRGref], on defining main DLT concepts; we refer the reader to those references for more details. We focus our brief introduction here on those concepts most important to understand from a communication perspective.

The core abstraction used in a DLT is that of a 'transaction', i.e., a cryptographically signed (set of) instruction(s) to modify a state machine, which in turn represents the distributed consensus the DLT is trying to maintain. These transactions are executed within the higher-level concept of a 'smart contract', which implements the specific DLT application, such as for cryptocurrency, storage management, decentralized governance, among others.

Valid transactions are maintained in a distributed 'ledger' in the form of hashed information referred to as 'blocks'. Consensus is represented through the longest available chain of blocks that can be obtained from another DLT peer.

The validation of transactions, and therefore the inclusion into the (distributed) ledger, is realized through the consensus layer, realizing computational operations, such as Proof-of-Work, Proof-of-Stake, and others. There has been much discussion on the implications of those computational aspects, e.g., on energy consumption, which is not the focus of this draft.

Figure 1 provides an overview of a typical layering within a DLT architecture. The focus of this draft is on the layers below the session, i.e. the communication that needs to be upheld in order to facilitate transactions and block exchange within the DLT system.

```
+------------+|-------------------------------------------------------+
| Application|| User     | DLT    | DLT     |  DLT    |Decentralized|
|  Layer     || Interface | Wallet | Explorer | Analytics |  Finance    |
+------------+|-------------------------------------------------------+
|App Protocol|| Identity | Token  | Storage |  DLT    |Decentralized|
|  Layer     ||  Mgmt    | Mgmt   |  Mgmt   | Oracle  |  Governance |
+------------+|----------------------------+---------------------------+
| Contract   ||        Transaction          |          Smart            |
|  Layer     ||          Engine             |        Contract           |
+------------+|----------------------------+---------------------------+
| Consensus  ||                PoW/PoS/DPoS/PBFT/Raft/etc.              |
|  Layer     ||                                                         |
+------------+|-------------------+------------------+------------------+
| Session    ||    Transaction    |       Block      |      Account     |
|  Layer     ||                   |                  |                  |
+------------+|-------------------+------------------+------------------+
| Transport  ||       TCP         |       QUIC       |       UDP        |
|  Layer     ||     (+TLS)        |                  |                  |
+------------+|-------------------+------------------+------------------+
| Network    ||    (DNS + ) IP    |     Service      |     Pub/sub      |
|  Layer     ||                   |     Routing      |     overlay      |
+------------+|-------------------+------------------+------------------+
| Resource   ||       CPU         |     Storage      |    Transport     |
|  Layer     ||                   |                  |     Network      |
+------------+|-------------------+------------------+------------------+
```

Figure 1: DLT Conceptual Architecture [IIC_whitepaper]

4. Communication in a DLT

With our focus on the communication impact of DLTs, we now tease apart the communication as it usually takes place in a DLT in order to realize the transactions within a distributed ledger and the maintenance of the latter. We first outline the interactions at a higher level before delving into the communication patterns that result from those.

As stated in the introduction, these insights are currently limited to those obtained from Ethereum, a proof-of-work based DLT platform. Future draft revisions will enrich this section with any differing insights from other DLT realizations and platforms.

4.1. DLT Interactions

We can distinguish three core interactions in a DLT:

1. A client commits a transaction to the DLT. The transaction request is being diffused across a set of DLT miners, which respond to the transaction request separately and add the transaction to their internal ledger information. The commit of the transaction leads to the miners committing compute and storage resources in relation to the smart contract that underlies the transaction. For this, so-called 'proofs' will be executed as part of the computational part of the DLT, although some methods for proof require additional communication to take place, e.g., election protocols.

2. The result of the aforementioned proof is a 'block' (of ledger information) that the miners in turn commit to a set of (other) DLT miners, which each receiving miner adds to their internal blockchain.

3. A client may query the latest blockchain, again from a set of miners to which the query is being sent. The longest returned blockchain represents the most trustworthy ledger information available.

We can see from those interactions above that communication in a DLT is multipoint in nature, i.e., transactions or information (such as blocks) are sent to a set of DLT peers, not just a single one.

Important here is that the set of DLT peers is a randomized sample from a larger pool of available DLT peers; this is to achieve diffusion among many DLT peers, avoiding repeated communication with a fixed set of DLT peers and thereby reducing the threat of collusion of information through a malicious set of DLT peers.

The consequence of that varying random nature of the multipoint diffusion, however, is that repeated unicast replication is utilized instead of efficient network-level multicast; this constitutes a first recognizable impact on provider networks.

In the following subsection, we now focus on the communication patterns that are utilized to achieve the aforementioned interaction. Special attention is here given to the establishment of the pool of DLT peers that is used in the multipoint operations that are executed for each interaction, be it a transaction or the commitment of a newfound (ledger) block.

4.2. Resulting Communication Patterns

As mentioned before, it is key for any DLT peer, be it a client or a miner, to establish and maintain a 'pool of peers' from which it can select a set of DLT peers for each intended interaction. Figure 2 outlines those steps, detailed in the following. Our insights on realization were obtained from an Ethereum based experiment, using the go-ethereum release V1.10.2-stable on a Linux-based machine, operating out of Munich, Germany.

1. The first phase is that of a 'peer discovery'. For this, an initial list of DLT peer information is obtained from a 'bootstrap node', of which only few exist in the DLT, holding the IP address and port information of each DLT peer that has signed up to the DLT overlay (other information may include DLT-specific information, such as an overlay ID or similar).

2. This initial list of DLT peers is now contacted through a (UDP-level) PING/PONG sequence, thereby discovering those DLT peers that are reachable for the DLT interactions.

3. A successful discovery of the DLT peer is now followed with the establishment of suitable transport security. Once successfully secured, the discovered DLT peer is being added to the 'DLT pool' list at the initiating DLT peer.

4. Once security is established, capabilities are exchanged that ensure that the discovered peer can successfully complete possible requests. Those capabilities may include HW capabilities (e.g., GPU usage, certain memory build-out), SW capabilities (use of certain hash functions, blockchain checkpoint) and others.

5. The initiating DLT peer repeats now the previous steps 1 through 4 until the pool size reaches a defined limit. Unlike contacting the bootstrap nodes, however, the newly and successfully discovered DLT peers in the previous round are contacted instead for obtaining a list of DLT peers.

6. Any member of the DLT pool is continuously checked for connectivity through frequent (e.g., TCP-based) HELLO messages. Any failed HELLO transaction leads to removing the DLT peer from the pool and obtaining another DLT peer as replacement.

The final size of the pool is a matter of local configuration (in our case about 28k DLT peers, significantly less than the size of the overall DLT network, which was about 500k at the time of the experiment).

Also, a DLT client may commence with transactions (to the DLT overlay) already while the pool creation is still ongoing, thereby progressing to the last step in Figure 2 once a suitable set of DLT peers can be obtained from the overall (and possibly still growing) local pool of peers.

```
+-------------------+                                if DLT peer connection fai
led
|    Obtain list    |<-------------------------------------+
|   of DLT peers    |<--+                                  |
+-------------------+   | if pool size       +--------------+---
|       Node        |   | smaller than max   |  Maintain peer  |
|      discovery    |   |                    |  connectivity   |
+-------------------+   |                    +-----------------+
|     Transport     |   |
|      security     |   |
+-------------------+   |
|    Capability     +---+
|     exchange      |
+-------------------+
          |
          |   add discovered peers to pool of DLT peers
         \|/
+--------------------------------+
|     Obtain set of DLT peers    |
|     from pool of DLT peers     |
+--------------------------------+
|         Transactions           |
+--------------------------------+
```

Figure 2: Steps of Communications in a DLT

5. Challenges for Users and Provider Networks

Considering the observed communication patterns in the previous section, we can identify a number of challenges that need addressing:

1. Reachability information is required to interact with other peers. For that, bootstrap nodes maintain IP addresses of all peers (plus port information). As illustrated in Figure 2, new DLT peers need to download and expand suitable reachability information upon joining, either from bootstrap node or via discovered nodes - see Figure 2, , requiring each DLT peer to maintain a pool of peer as active connections.

2. Clients know nothing about capabilities of peers to serve requests. In other words, the discovery in Figure 2 merely ensures possible reachability but not necessarily successful communication. As a consequence, the resulting approach, illustrated in Figure 2, is to (1) contact potential peer, (2) wait for connection, (3) inquire capabilities, (4) disconnect if not matching. Here, peers may never reply to connection establishment (step 2), usually resulting in additional latency due to timeouts involved, prolonging therefore the establishment of the pool of peers to communicate with. Such capabilities often reflect the continuous evolution of business models over DLT networks and may be dynamic in nature. For example, the minimum transaction fee may depend on the 'DLT gas price', which is set up at the transaction recipient (miner).

3. Peers map sending of transactions onto unicast communication, which negatively impacts efficiency (bandwidth usage) and transaction completion time. Here, the use of group-based multicast approaches is difficult due to the random nature of the set of peers selected for communication in every request exchange, aiming at the diffusion of requests rather than interacting with a stable (but possibly colluding) set of peers.

4. DLT peers need to expose their IP address to the DLT system, replicated to the bootstrap nodes, but also other DLT peers by virtue of the discovery process outlined in Figure 2. This may lead to privacy and/or security issues in the form of geo-identifying specific peers, DoS attacks on particular parts of the DLT and others.

6. Experimental Insights

To shed some more light onto the possible impact on provider networks, stemming from some of the challenges in Section 5, we conducted experiments, using the same setup described in Section 4.2. More details (and suitable graphical representations of our initial results can be found in [IIC_whitepaper]).

Here, the goal was to undergo the steps needed to build up the needed pool of DLT peers, after which we sought to synchronize to determine the longest blockchain available in the discovered pool. The resulting geographic spread of the discovered DLT peers included all continents albeit with an expected clustering of nodes North America, Europe, Asia, and Australia, with only few discovered in South America and Africa.

6.1. Types of DLT Peers

Our first target was to differentiate types of DLT peers that stem from the communication patterns in Figure 2. Specifically, we came to differentiate the following types of DLT peers:

1. Non routable peers: This type include all those peers that never positively responded to step 1 of the discovery, i.e. the PING/PONG to determine reachability. Reasons here may include to be located behind a firewall, being intermittently available (and switched off during the connection attempt), or simply having left the DLT while still remaining in the information pool maintained at the bootstrap nodes.

2. Signalling peers: This type includes peers that respond positively to reachability but do not positively succeed in the transport security or capability exchange steps (blockchain checkpoint).

3. Dropped data peers: This type of peers successfully complete all discovery steps, thereby end up in the pool of peers, but still do not provide suitable data upon request (here a valid blockchain information). The reasons here could be unavailable information or not completing the transfer of information (blockchain information can be very large, several GBs, so that DLT peers may run out of available BW budget or decide to sever the connection because of switch-off or other reasons during the transfer). While here communication in the DLT does take place, it is not successful in regards to the intended communication, therefore wasted.

4. Data peers: This final type of peers successfully completes all steps in Figure 2, i.e. not only the discovery but also the intended transfer of DLT-relevant data.

In our experiments, we determined at about 18% of peers are of the last type, i.e. successfully contribute to DLT purposes, while about 2% are of the third category, about 12% are non routable peers and about 68% are signalling peers. In other words, almost 80% of all attempted discoveries fails either because of the lack of reachability or mismatching capabilities.

6.2. Communication Waste

Looking at the bandwidth usage across the different peer types allows for shedding some light on the communication that needs to be carried through the participating provider networks.

Given the amount of data for each blockchain synchronization, it is not surprising that, despite forming a mere 18% of peers, the 'data peers' account for about 58% of traffic in the overall system. This is followed by the 'dropped data peers' with about 31.5% (since still much data is sent albeit unsuccessfully). Both non routable and signalling peers account for a total of slightly under 10% of data used.

Although the amount of data that is wasted here accounts for (significant) total of about 42%, the data-heavy operation of synchronization large amounts of (blockchain) data is mainly to blame for this; however, the synchronization has to happen for any DLT peer to start operating as a possible DLT miner, so is not avoidable.

7. Opportunities for Network Innovations

The challenges outlined in Section 5 lead us to outline possible opportunities for network innovations that may address those challenges and reduce the observed impact on provider networks. We stress here that none of the suggested approaches constitute solutions for those opportunities but merely possible starting points beyond which further study is required:

1. Addressing model: With the DLT overlay being realized over an IP network, each DLT peer is being addressed via its IP(v4/v6) address. With the discovery step selecting a dedicated DLT peer (through its IP address), the discovery steps (see Figure 2) include dedicated steps to ensure the reachability of the specific DLT peer under discovery. Until reachability can be ensured, traffic (in the form of PING/PONG messages) and latency (through sending those messages, while needing to wait for a timeout in case the DLT peer is not routable) need to occur, despite the DLT peer not being eventually used for communication.

    * Approaches such as those in [SOI][SarNet2021][IFIPNetworking2022] may allow for DLT peers to advertise their capability to serve as a miner by using 'service announcements' that expose the capability to serve transaction requests, which each announced DLT peer representing a service instance of the announced mining service. Such native L3 (or L3.5) level service routing capability would therefore remove any of the discovery steps and the maintenance of the dedicated DLT overlay infrastructure. Furthermore, it would remove any visibility of individual DLT peers' reachability information from other miners, until directly communicating with a specific DLT peer (for which the peer's IP address may be used, as suggested in [SarNet2021][IFIPNetworking2022]). Last but not least, being

able to send a request without previously forming a pool of DLT peers (which is smaller than the number of all DLT peers in the overlay) also increases the possible number of DLT peers to communicate with rather than being limited to the peer-specific pool.

2. Constraint-based peer selection: Following on the aspect of relying purely on reachability information in the form of IP addresses, the discovery steps in Figure 2 further include a number of capability-dependent selection criteria to finally include a DLT peer in its pool of peers. Specifically, the security and capability exchange may lead to a disconnect from a successfully contacted DLT because of such exchange leading to mismatching capabilities. Furthermore, even after an initial capability exchange being successful, the actual transaction itself may be constrained by capabilities such as available resources (e.g., bandwidth or CPU), leading to unsuccessful communication, which in turn will need to be compensated with including another DLT peer into the diffusion request.

   * Approaches such as [SarNet2021][IFIPNetworking2022] may allow to constrain the forwarding to one of possible many DLT peers. Hence, the capabilities used in the current DLT steps Figure 2 could be encoded as suitable constraints for such selection, the constraints itself being advertised as part of the service announcement (see above). As a result, the request will be forwarded to those destinations only which have previously announced constraints that match those of the request, thereby ensuring the successful completion of the request - further study is needed for those situations in which constraints may change frequently, thereby leading to successful matching, yet still unsuccessful request completion.

3. Diffusion multicast: The multipoint replication of the transaction request to a set of DLT peers, chosen from the larger DLT pool maintained at the initiating DLT peer, increases the overall system but, in particular, individual client bandwidth usage, which in turn impacts the provider network by needing to provide the necessary resources for the replicated sending.

   * Approaches such as those in [SOI][SarNet2021][IFIPNetworking2022] may allow for sending a service request to a given number of DLT peers, where the replication is part of the constraint-based forwarding decision, thereby optimizing the packet delivery through in-network instead of endpoint-based replication. The challenge here lies in preserving the diffusion character of the multipoint operation. In other words, the set of DLT peers

used for the transactions changes for each request with a randomization that attempts to prevent possible collusion through DLT peers. With that, typical group-based methods, most notably IP multicast, do not suffice.

8. Relation to IETF/IRTF and IEEE SA Efforts

Both, DLTs as well as routing innovations, are subject to investigation in a number of related IETF and IRTF efforts. For instance, the Decentralized Internet Infrastructure RG [DINRGref] has been studying various aspects of DLTs and blockchains. Our findings in this draft may provide additional input into the work of this RG, while we would solicit feedback from this group of experts into the specific insights we have derived so far.

There is no standard way of providing interoperability between DLT networks. This results in difficulty of transferring or exchanging virtual assets from one DLT network to another. An interoperability architecture is being proposed in the IETF [I-D.hardjono-blockchain-interop-arch] to permit two gateways, belonging to distinct DLT networks, to conduct a virtual asset transfer between them while ensuring the asset does not exist simultaneously on both networks. The Open Digital Asset Protocol (ODAP) [I-D.hargreaves-odap] is a gateway-to-gateway protocol to perform a unidirectional transfer of a virtual asset.

Blockchain technologies, and thereby DLTs, have also been proposed for use in network functions itself. For instance, the work in [I-D.mcbride-rtgwg-bgp-blockchain] proposes to position BGP as DLT-managed transactions and thus, to utilize the power of a permissionless (DLT-based) management infrastructure to improve on resilience and trust into the operations performed within BGP, such as origin announcements and BGP updates. Such proposition, however, opens the question on the exact nature of such infrastructure but also its impact in terms of incurred traffic, particularly when operating at scale.

Furthermore, routing innovations under the label of 'semantic routing' have been the topic of recent work, see [I-D.farrel-irtf-introduction-to-semantic-routing] for an overview. With the examples of service routing as possible approaches to realize the opportunities outlined in the previous subsection, a stronger linkage to this activity should be considered.

While the DLT standardization efforts in IEEE SA mainly focus on the upper layers of the DLT architecture, the decentralized identity related standards (e.g., P2958 [P2958] and P3210 [P3210]) that are currently under development might be relevant for addressing specific challenges in the DLT network layer.

9. Open Questions

The work initially presented in [IIC_whitepaper] focused on the specific impact that DLT operations may have on provider networks, thereby turning the attention not to the specific applications of DLT but what their realization may mean to the underlying network operators.

Although attempting from the onset to base our insights on actual experiments we conducted, we recognize that those insights are only the start to a possibly wider understanding beyond this initial work.

We therefore solicit not only feedback on the specific findings presented in the previous sections, but also to specific questions that our work has led to:

1. Correctness of observed DLT behaviour: Is our observed behaviour correct or have we overlooked important aspects?

2. Depth of insights: Can we deepen our insights through more experiments, focus on different or more KPIs?

3. Transfer of insights: Our insights so far are based on the Ethereum DLT system. How transferable are the observed patterns of communication onto other DLT systems that are in use?

4. Differences in DLT realizations: If the answer to the previous question leads to little transfer onto other DLT platform, can we distil those difference with the goal to develop a wider methodology to capture DLT behaviour?

5. Applicability of other network innovations: What other network innovations may address the specific impacts we identified in our study? Which ones beyond the ones currently listed should be included?

10. Next Steps

As for the next steps for this draft, the authors seek to deepen the current insights through further conducted experiments, providing more insights on the disconnects experienced by the system and the costs for maintaining the pool of DLT peers.

Furthermore, the authors will more directly link to relevant network innovations, particularly in the service routing and instantaneous multicast domain, with the goal of providing estimates of improving on the operational costs of DLTs through such new network innovations.

11. Conclusions

This draft is a living document, originating from an initial study in the impact of DLTs on provider networks [IIC_whitepaper].

As such, the authors solicit feedback from the wider DLT and network community to improve on the insights, transfer them onto more DLT systems, and shed light onto how possible network innovations could improve on the identified issues.

12. Security Considerations

This document does not introduce or modify any security mechanisms. The nature of DLTs is to provide a high level of transactional security through immutability of the data in blocks. But 51% attacks are possible amongst miners particularly on smaller, private blockchains where legitimate miners could be prevented from completing blocks and new blocks could be created by illegitimate miners. Smart contracts could become vulnerable if a function calls the wrong contract either intentionally or through human error. Transactional data meant to be private might be exposed. DLT attacks most often involve accounts being hacked outside of the DLT domain.

13. Privacy Considerations

Since the IP addresses of DLT peers are exposed in the DLT system, the DLT network layer might be subject to privacy leakage. This document does not introduce any mechanisms for protecting IP address privacy and the methods described in [I-D.ip-address-privacy-considerations] could be employed to enhance the privacy of DLT peers.

14. IANA Considerations

This draft does not request any IANA action.

15. Acknowledgements

This draft acknowledges the work done in the IIC Industrial Digital Ledger focus group, leading to the whitepaper in [IIC_whitepaper]. We would like to thank the co-authors of this whitepaper for their work, specifically David Guzman (Huawei Technologies), Abhijeet Kelkar (GEOOWN Consulting), Xinxin Fan (IoTex), Mike McBride (Futurewei Technologies), Lei Zhang (iExec), Ulrich Graf (Huawei Technologies) and Dirk Trossen (Huawei Technologies) but also Stephen Mellor (IIC staff) who oversaw the process of organizing the contributions.

16. Informative References

[DINRGref] "Decentralized Internet Infrastructure (dinrg)", WG DIN Research Group, <https://irtf.org/dinrg>.

[DLT_intro] Antonopoulos, A. M., "Mastering Bitcoin, 2nd Edition", Book O'Reilly Media, Inc., 2017, <https://www.iiconsortium.org/pdf/2022-01-10-Impact-of-Distributed-Ledgers-on-Provider-Networks.pdf>.

[DLT_intro2] Rauchs, M., Glidden, A., Gordon, B., Pieters, G., Recanatini, M., Rostand, F., Vagneur, K., and B. Zhang, "Distributed Ledger Technology Systems: A Conceptual Framework", Report Cambridge Centre for Alternative Finance, 2017, <https://www.jbs.cam.ac.uk/wp-content/uploads/2020/08/2018-10-26-conceptualising-dlt-systems.pdf>.

[I-D.farrel-irtf-introduction-to-semantic-routing] Farrel, A. and D. King, "An Introduction to Semantic Routing", Work in Progress, Internet-Draft, draft-farrel-irtf-introduction-to-semantic-routing-04, 25 April 2022, <https://datatracker.ietf.org/doc/html/draft-farrel-irtf-introduction-to-semantic-routing-04>.

[I-D.hardjono-blockchain-interop-arch] Hardjono, T., Hargreaves, M., Smith, N., and V. Ramakrishna, "Interoperability Architecture for DLT Gateways", Work in Progress, Internet-Draft, draft-hardjono-blockchain-interop-arch-03, 6 November 2021, <https://datatracker.ietf.org/doc/html/draft-hardjono-blockchain-interop-arch-03>.

[I-D.hargreaves-odap]
Hargreaves, M., Hardjono, T., and R. Belchior, "Open Digital Asset Protocol", Work in Progress, Internet-Draft, draft-hargreaves-odap-03, 6 November 2021, <https://datatracker.ietf.org/doc/html/draft-hargreaves-odap-03>.

[I-D.ip-address-privacy-considerations]
Finkel, M., Lassey, B., Iannone, L., and B. Chen, "IP Address Privacy Considerations", Work in Progress, Internet-Draft, draft-ip-address-privacy-considerations-03, 10 January 2022, <https://datatracker.ietf.org/doc/html/draft-ip-address-privacy-considerations-03>.

[I-D.mcbride-rtgwg-bgp-blockchain]
McBride, M., Trossen, D., Guzman, D., and T. Martin, "BGP Blockchain", Work in Progress, Internet-Draft, draft-mcbride-rtgwg-bgp-blockchain-03, 6 February 2024, <https://datatracker.ietf.org/doc/html/draft-mcbride-rtgwg-bgp-blockchain-03>.

[IFIPNetworking2022]
Khandaker, K. S., Trossen, D., Khalili, R., Despotovic, Z., Hecker, A., and G. Carle, "CArDS: Dealing a New Hand in Reducing Service Request Completion Times", Paper IFIP Networking, 2022.

[IIC_whitepaper]
Trossen, D., Guzman, D., Kelkar, A., Fan, X., McBride, M., Zhang, L., and U. Graf, "Impact of Distributed Ledgers on Provider Networks", Whitepaper Industry IoT Consortium Whitepaper, 2022, <https://www.iiconsortium.org/pdf/2022-01-10-Impact-of-Distributed-Ledgers-on-Provider-Networks.pdf>.

[P2958] "P2958: Standard for a Decentralized Identity and Access Management Framework for Internet of Things", Standard IEEE Standards Association., <https://standards.ieee.org/ieee/2958/10483/>.

[P3210] "P3210: Standard for Blockchain-based Digital Identity System Framework", Standard IEEE Standards Association., <https://standards.ieee.org/ieee/3210/10242/>.

[SarNet2021]
Glebke, R., Trossen, D., Kunze, I., Lou, Z., Rueth, J., Stoffers, M., and K. Wehrle, "Service-based Forwarding via Programmable Dataplanes", Paper 1st Intl Workshop on Semantic Addressing and Routing for Future Networks, 2021.

[SOI] Jiang, S., Li, G., and B. Carpenter, "A New Approach to a Service Oriented Internet Protocol", Paper IEEE INFOCOM 2020 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), 2020.

Authors' Addresses

Dirk Trossen
Huawei Technologies
Munich
Germany
Email: dirk.trossen@huawei.com

David Guzman
Technical University Munich
Munich
Germany
Email: david.a.guzman.c@gmail.com

Mike Mc Bride
FutureWei
Email: michael.mcbride@futurewei.com

Xinxin Fan
IoTeX
Email: xinxin@iotex.io

Network Working Group
Internet-Draft
Intended status: Informational
Expires: 1 January 2023

D. Trossen
D. Lou
S. Jiang
Huawei Technologies

30 June 2022

Continuing to Evolve Internet Routing Beyond 'Mere' Reachability
draft-trossen-rtgwg-routing-beyond-reachability-01

Abstract

This document discusses the evolution of the Internet routing system beyond mere reachability. We observe, through examples of past development, that such evolution has been taking place to improve on capabilities of the Internet, deal with more complicated network deployments and cater to changing requirements by end users as well as novel and emerging applications.

For achieving a routing system that serves more than a singular reachability purpose, more information is taken into account when performing the purpose-specific functions. Such extra information can be obtained by extending current routing protocols to exchange more information or by carrying that information within packets.

This document is intended to seed discussions of how the observed evolution of the Internet's routing system can continue, what issues may occur when simply continuing the current approach for achieving routing beyond 'mere' reachability and what may be needed to address those issues. Ultimately, however, this document recognizes the positive impact that moving beyond reachability has brought to the Internet and will continue to do so.

This Internet-Draft will expire on 1 January 2023.

Table of Contents

1. Introduction

The current routing system was initially designed for a single purpose - reachability. That is, it was built to find paths through the network so as to forward packets to their destinations. The routing system has successfully supported the Internet as it grew from a very small scale network to a giant system that covers the whole world with billions attached devices and users. This routing

system has done a good job for global reachability, however, through the years, many other needs or purposes have arisen in the Internet, such as hostname/address mapping, destination selection, security, privacy, group isolation, various QoS requirements etc. Many of these additional needs or purposes have resulted in the development of extended and distinct systems, such as DNS, patched firewall, DPI, and CDN, etc. These systems have worked well but with costs in terms of quality of experience for the user, particularly with respect to time delay, but also with respect to costs of development, deployment and management throughout (parts of) the Internet.

An alternative approach is the integration of extra capabilities and purposes into the routing system directly. By exchanging necessary additional information or including such information in the packet header, purposes beyond just reachability have found entry into the routing system over the many years of the Internet's development.

This document presents a brief survey of solutions that, when combined, represent a routing system beyond reachability that effectively forms today's Internet. While this survey somewhat relates to that presented in [I-D.king-irtf-semantic-routing-survey], our focus here is on the identification of the underlying purpose for developing extensions, not on the body of work that represents an approach for doing so (named 'semantic routing' in the above draft). However, [I-D.king-irtf-semantic-routing-survey] may be useful for more information on the specific extensions.

Some of these extensions are intended to be deployed in limited domains [RFC8799], while others are intended for use across the Internet. The boundary of limited domains may also be the boundary of purposes and semantics of information defining those purposes. This survey is used to demonstrate the recognized need by those having developed existing solutions for the Internet's routing system to have multiple purposes beyond mere reachability.

Building on the survey and our summary, we recognize that, in many parts, the Internet has already evolved into a 'multi-purpose routing' system. However, we identify issues with the approach that has been taken so far, namely that of purpose-specific extensions. We assert that these issues will increasingly impede the Internet's ability to accommodate future purposes (represented in the form of new use cases), if we simply continue with a 'business as usual' attitude towards developing purpose-specific solutions for them.

Instead, we position this document as the starting point for a discussion on how to evolve the Internet routing system in a coherent manner that will help us avoid the identified issues outlined in Section 4, while still allowing for integrating evolving the

semantics of communication along the lines outlined in Section 5 towards new purposes for Internet routing as they will emerge in the future.

Note: This document does not discuss how routers may use policies, that are coded in, configured at, or signaled to it, to make routing decisions. It does neither pass comment on the advisability or practicality of any of the proposals nor does it define any technical solutions.

2. Reachability - Original Purpose of the Routing System

Network routing protocols were initially designed to enable forwarding of IP packets through the network toward destination addresses. Fundamental to this is the locator semantic of IP addresses, which has been assigned in the context of network topologies. The original routing system was designed on a distributed basis. Each router makes its own decision about the interface/link onto which it forwards a packet. Each decision takes the packet one hop closer to the destination. However, the choices made by distributed devices may not always work well if they are poorly coordinated between the routers, resulting in issues, such as forwarding loops, which may be transitory or permanent. So it is normal to require the use of the same algorithm to decide the forwarding actions at each hop.

A way to avoid routing issues is to select an end-to-end path a priori and consistently execute forwarding on the intermediate routers accordingly. This element of traffic engineering is known as "path steering" [I-D.ietf-teas-rfc3272bis] and relies on the routing to protocols collect and exchange the reachability within a domain, so that any routers can select an end-to-end path . However, the amount of information needed to support these decisions can become very large (e.g., in large networks, with many possible paths and route metrics), which might impede the scalability both in terms of the storage and the distribution of the information. Although network topology filters are often applied to reduce the storage of the network data and the complexity of the computation algorithm, the path computation accuracy and optimality may be negatively impacted.

The Internet is a very complicated system that is made up of many independently built networks with two types of routing protocols: an interior gateway protocol (IGP) that routes inside a network and an exterior gateway protocol (such as BGP) that routes between networks. For a communication that crosses more than one domain, there could be many possible paths for the given destination. In principle, the more information that decision-making devices have, the better choices they can make. However, it is often infeasible to have all

information of all potential end-to-end paths, particularly for communications through several networks with different ownership. Consequently, the best choices made within each domain may not reach the best overall result. A key challenge here is the tussle between abstraction, needed for scalability, and optimality, which abstraction may impede.

When choosing the best paths or topology structures, the following may need to be considered:

* The method by which a path, or set of paths, is to be calculated. For example, a path may be selected automatically by the routing protocol or may be imposed (perhaps for traffic engineering reasons) by a central controller or management entity.

* The criteria used for selecting the best path. For example, classic route preference, or administrative policies such as economic costs, resilience, security, and (if requested) applying geopolitical considerations.

3. Extension of the Routing System Beyond 'Mere' Reachability

In the following, we provide a brief overview of routing extensions with purposes beyond 'mere' reachability. We align our overview with many of the solutions described in [I-D.king-irtf-semantic-routing-survey] and refer to this draft for more detail, in addition to the example references themselves.

The following Table 1 focusses on three key aspects when considering routing extensions for our discussion in this draft:

* Purpose: What is the intended purpose of the proposed extension? This aspect may lead to a taxonomy for looking at the capabilities of a multi-purpose routing system.

* Approach: What is the underlying technical approach to achieve the intended purpose? This aspect may lead to a taxonomy of approaches for achieving desired routing purposes.

* Examples: What are known examples that have employed the given approach to achieve the given routing purpose? This aspect provides a possibly growing catalogue of explicit examples to study in more detail.

Purpose	Approach	Examples
Path Selection for Traffic Engineering	Preferential Routing	IS-IS Extensions [RFC5305]
	Policy-based Routing	PBR models [RFC1104] Inter-domain policy routing [RFC1479]
	Flow Steering	TBD
	Path Computation	PCE [RFC4655] PCEP [RFC5440] PCEP Extension [RFC8231]
	IRTF	Path-aware Networking RG [PANRGref] Path properties [I-D.irtf-panrg-path-properties] Past efforts evaluation [I-D.irtf-panrg-what-not-to-do]
Path Selection for Multicast	Multicast	IP multicast [RFC1112] IPv6 addressing [RFC4291] MBone [MBONEref] MADCAP [RFC2730] MALLOC [RFC6308] MASC [RFC2909] MZAP [RFC2776] MSDP [RFC3618] SSM [RFC4607]
	Automatic Multicast Tunneling	AMT [RFC7450]
	Path-based Forwarding	BIER [RFC8279] BIER encapsulation [RFC8296] IP-over-ICN [I-D.trossen-icnrg-internet-icn-5glan]
Routing Architectures	Future architectures	[RESEARCHFIAref] [ITUNET2030ref] [SCIONref] [RINAref]
Service Function Chaining	L2/L3 Explicit Header Chaining	SFC [RFC7665] NSH [RFC8300] MPLS encapsulation [RFC8596]
	Name-based Chaining	Name-based SFF [RFC8677]
	Source Routing	Segment Routing [RFC8402]
Application/	Application-	Aalto [RFC7285] APN

service-aware Routing	server based	[I-D.li-apn-framework]
	L3 based	Dyncast use cases [I-D.liu-dyncast-ps-usecases] Dyncast use architecture [I-D.li-dyncast-architecture]
	Network programming	Segment routing [RFC8986]
Anycast Routing	IP Anycast	Architcture considerations[RFC7094] Operation of Anycast [RFC4786]
	Metric-based	Dyncast use cases [I-D.liu-dyncast-ps-usecases] Dyncast architecture [I-D.li-dyncast-architecture] Load-balanced anycast [weightedRef]
Privacy-aware Routing	Crypto routing	CGA [RFC3972] CGA Extension Field [RFC4581]
	Obfuscation	ILNP-based [ILNP_PRIVACY]
Security-enhanced Routing	Routing Architecture	SCION [SCIONref]
Identity Split Routing	Identity/ Locator Split	LISP [RFC6830] LISPbis [I-D.ietf-lisp-rfc6830bis] LISPbis [I-D.ietf-lisp-rfc6833bis] HIP[RFC4423] ILNP [RFC6740]
Content Routing	Routing over content names	ICN [ICNref] NDN [NDNref] ICN deployment [RFC8763] HICN [HICNref]
	Routing via indirection points	DNS DONA [DONAref]
Differentiated Routing	QoS Differentiation	DiffServ [RFC2474] IntServ [RFC2210]
	Path differentiation	Segment Routing [RFC8402] SFC [RFC7665]
Extended	EH-based	IPv6 EH [RFC8200]

Routing		

Table 1: Summary of Routing Extensions

4. Issues with Current Approaches

Developing routing purposes beyond the original 'mere' reachability does come with issues when considering their deployment and use in the Internet; we outline those issues in the following.

We note that those issues are intrinsically linked to the ones stemming from the extension of addressing semantics that may be used to realize the various routing extensions, identified in [I-D.jia-intarea-internet-addressing-gap-analysis]. We therefore structure our presentation along the same lines.

4.1. Limiting Routing Semantics

Approaches that intend to change the purpose of communication, specifically within the evolution of communication semantics outlined in Section 5 through, e.g., by separating host from network node identification [RFC7401] or through identification of content directly [HICNref], are limited by the reachability purpose of IPv6, as defined by its source and destination address.

This leads to approaches such as [I-D.trossen-icnrg-internet-icn-5glan] to override addressing semantics, namely replacing the IPv6 source and destination addresses with path information instead, in order to achieve the desired purpose of its routing solution. This, in turn, requires to still carry address information as part of the payload in order to support clients unaware of the routing extension. Furthermore, such approach may lead to 'information leakage'outside the boundaries of the system in which its changed purpose is being realized. Introduction of dedicated gateways to 'translate' from one purpose (new routing) to another (IPv6 routing) is the consequence of this.

But even such approach of 're-writing' packet information towards a new purpose limits the expressible (new) semantic information to the size of the original field, thereby limiting the support of content information in approaches such as [HICNref] or the size of supported networks in [I-D.trossen-icnrg-internet-icn-5glan] to the bit size afforded by IPv6 addresses.

4.2. Complexity and Efficiency

Introducing new routing purposes also bring additional complexity. This becomes an issue when new purposes are being introduced in particular parts of the overall Internet, such as the edge of the network, while relying on the existing reachability purpose of the Internet to interconnect those islands over the existing Internet.

This additional complexity therefore often comes with a penalty in the form of efficiency and costs for realizing the novel routing purpose, which in turn may specifically pose an even bigger problem when the solution is introduced at the edge of the network, which is often constrained in resources and therefore costs that can be expensed.

For instance, if the specific new purpose requires compression of packet fields, such as for header compression, additional processing as well as potentially required gateways that restore information towards the Internet may be prohibitive for introducing the desired new routing purpose in this part of the Internet.

Conversely, performance requirements of core networks, in terms of packet processing speed, pose a problem when wanting to accommodate novel routing purposes. Here, not only the possibly additional processing but also the required changes of often HW-based platforms makes adoption of novel routing purposes prohibitive.

4.2.1. Repetitive encapsulation

A routing solution targetting a different purposes often requires encapsulating the relevant information, thereby bloating packet sizes and lowering overall efficiency. This can be seen in routing solutions such as [I-D.trossen-icnrg-internet-icn-5glan] (introducing an alternative forwarding solution), [I-D.ietf-lisp-mn] (handling mobility in LISP), [RFC8926] and [RFC7348] (DC networking), [RFC8986] (traffic engineering) as well as [TOR] (routing privacy), all of which introduce multiple encapsulations that in turn reduce the forwarding effiency.

The introduction of dedicated points of encapsulation also introduce complexity and costs at the points of the network where they are required, which may often be at the network edge, while also establishing failure points and therefore increasing the overall fragility of the system; a point we discuss in more detail in Section 4.4.

4.2.2. Introducing Path Stretch

Path stretch is an issue when moving from direct reachability purposes to additional ones, such as dealing with mobility of endpoints, as done in MobileIP [RFC6275]. In this case, following the typical triangular route affects transmission effiency as well as overall latency of the communication, instead of communicating directly towards the (new) network location.

Additionally, the realization of novel purposes, such as privacy-compliant routing in systems like TOR [TOR], often introduce path stretch due to the additional relays being introduced for fulfilling the intended purpose, here the obfuscation of traffic for privacy reasons.

4.2.3. Complicating Traffic Engineering

As outlined in Section 3, many solutions to extend the original reachability purpose of Internet routing aim to introduce or improve on traffic engineering capabilities, e.g., by enabling decisions based on QoS metrics, mobility, chaining and others aspects.

However, realizing each novel purpose as a separate solution in itself likely hampers the goal for which they are developed, namely to improve on traffic engineering, whenever individual solutions are being used in combination. This 'feature interaction' aspect may even prevent combined uses, while at a minimum requiring an understanding if combined uses are possible in the first place or instead incompatible with each other. This is not just an issue that routing purposes may be incompatible at a functional level, e.g., through conflicting policies, but may also utilize conflicting realizations for their purposes.

4.3. Security

Security issues, outside the security considerations of the specific design, often arise from the integration of the specific solution into the existing routing system. For instance, HIP [RFC7401] limits its host identity to 128bit in an effort to be backward compatible, but possibly resulting in weak cryptographical strength. A similar issue can be observed in CGA [RFC3972], where only 59bits of the 128bit limit may be used for what could be packet signatures not sufficiently robust enough against attacks.

Attempts to introduce privacy purposes into the routing system, e.g., by utilizing ephemeral addresses [I-D.gont-v6ops-ipv6-addressing-considerations], may in turn pose significant challenges on the routing system through its required renewal rate of addresses.

4.4. Fragility

From the overview of novel routing purposes in Section 3, we can observe that the existence of those additional routing purpose adds many purpose-specific translation/adaptation points, responsible for mapping formats from one meaningful context into another. This is in turn creates dependency on this additional functionality to exist for endpoints to communicate with the context of the intended purpose.

While translation/adaptation between purposes and their defining contexts is often not avoidable when going beyond 'mere' reachabiulity, it is the solution-specific nature of those components (required for many if not each extended purpose) that is likely to increase the fragility of the resulting system.

The key problem here is the interaction with other extended purposes that may exist in specific deployments. While needing to operate in the presence of those other purpose-specific components, their design has often not taken into account the specific interaction in question. Given the diversity of extension realizations, utilizing many, almost any packet field, even beyond and entirely different to its intentded purpose, conflicting behaviour as well as diverging interpretatin of the utilized packet information is clearly an issue. Only careful testing of combinations with possible delineation (of purposes) as well as networks may be required, all of which further increases the costs for utilizing the extended purposes.

4.5. Interoperability

Although routing extensions are developed with their specific purposes in mind, reflected in requirements and behaviours, they are often realized in conjunction with other extensions when it comes to real-world deployments.

This poses an Interoperability challenge, both in terms of backward as well as forward compability. Feature interactions need investigations, often left to operational deployment.

Building extensions on the basis of agreed packet field semantics is one way to achieve the desired interoperability, unless approaches use extensions to packet fields beyond their original intention. As a consequence, translation/adaptation points may be needed to ensure interoperability with other parts of the network, increasing the fragility of the system, as discussed in Section 4.4.

Forward compability aims at ensuring that future extensions will continue to be possible, aiming at an overall extensibility of the system beyond its purpose at the time of developing a specific solution. IPv6 extension headers are one example of enabling future extensions, although not without their own problems in real-world deployments [SHIMv6ref].

5. The Driving Need for Evolving Communication Semantics

When looking at the evolution of routing beyond reachability, the key question arises on how the purposes of communication, or more concretely the underlying communication semantics, have evolved from the shortest-path routing of packet from sender to a receiver, each of those being originally identified through IP locators and captured as a source and destination address field in packet headers but having evolved through approaches such as those presented in Section 3.

To better understand this evolution, we distinguish communication in networks according to the relationship between senders and receivers and the selection of the paths and endpoints for the delivery of packets, leading us to the following distinct semantics.

* The Unicast semantic consists of sending a packet from a sender to a single receiver.

* In Anycast, a packet is sent from the sender to any one of a set of receivers, where the choice of receiver is made by the network.

* In Multicast, a packet is sent from a sender to all members of a group of receivers.

The identification of endpoints in these semantics may use well-known IP locators for unicast relations or IP multicast groups, while Anycast may use an IP anycast address or a content/service name [NDNref][CARDS]. Often, packets also carry higher-layer information, such as ports, to facilitate the endpoint-local handling of received packets.

These relationship semantics can be further constrained through path and endpoint selection semantics:

* Multicast relations may be defined as (i) by configuration, (ii) dynamically formed through a membership protocol [RFC3376], (iii) through requests towards the sender [I-D.trossen-bier-frrm], or (iv) through diffusing towards a sub-group of a larger group, e.g., in Distributed Ledger Technologies (DLTs).

* In Bestcast, the network applies constraints to determine the best path to the receiver based on the destination address, the state of the network and the compute resources, and information supplied with the packet. Bestcast may also be achieved by extending the anycast address to include multiple virtual unicast representations of the same receiver. The choice of a specific receiver may also determine the network path to reach this receiver. The choice may be made within the network or using a server-based scheduler and a database akin to DNS Resource Records.

* The Chaincast semantic steers a packet through a specific set of nodes deduced from the value of the destination address, with typical examples being Service Function Chaining [RFC7665] and Segment Routing Network Programming [RFC8986].

While we can see many examples of those evolving communication semantics, a crucial question is 'What are the things that are identified by the identifiers?' [RTGWGinterim]. Behind this question is the observation that 'if you want to put multiple definitions into the same identifier space, then it requires an architecture discussion.'

This interjection is key in understanding the architectural dimension of evolving communication semantics since those evolved meanings are often based on differently identifying the 'ends' of the communication. Information-centric networking (ICN) [NDNref] is one such example, turning the meaning of an address from being a network location into one where the address represents a piece of information, with the network being tasked to build ephemeral relations between those network components asking for the information and those that may be able to provide it.

The FRRM (forward request return multicast) [I-D.trossen-bier-frrm] semantic for multicast relations is another approach (albeit related to ICN), where the commonality of the forward requests, e.g., in the form of a URL pointing to the same content chunk, identifies the communication relation akin to ICN, while path information (e.g., in the form of BIER forwarding information [RFC8279]) is used to actually forward the packets from its source to the possible receivers.

Architectures, such as those for ICN and IP, have long lived in parallel, e.g., with ICN deployed in limited domains [RFC7665] or interconnecting to the Internet through dedicated application-level gateways, while proposals such as [HICNref] utilize in-address embedding to deploy ICN alongside IP networks.

The architectural question that arises from this is what the overarching architectural principles as well as its resulting frameworks and architectures should look like that would allow not only for rich communication semantics to be implemented but also to emerge over time and continued to be supported without resorting to gateway and in-lay techniques that all come with complexity and fragility issues?

6. Where to Go From Here?

This document outlined the original starting point of the Internet's routing system, namely providing 'mere' reachability, and showed through its survey of existing solutions that have since been developed that Internet routing has, in fact, evolved into a system that serves many purposes beyond its original 'mere' reachability goal.

However, the issues we outlined in Section 4 pose the question on how to move forward in the (future) evolution of Internet routing. We assert that continuing with a 'business as usual' attitude will ultimately compound the identified issues, thereby hampering innovation in novel routing purposes and solutions, and therefore the Internet overall.

As a way forward, we ask the wider RTG WG community to recognize the following cornerstones for an evolution path for Internet routing:

1. Further evolution of the Internet's routing system MUST take a wider architectural approach in order to break with the point solution approach that has led to the identified issues in Section 4.

2. With research and development on routing solutions continuing, as also illustrated in [I-D.king-irtf-semantic-routing-survey], these works MUST be brought into the process of IETF engagement and standardization to increase the understanding of what novel trends, works, and possible developments may be just around the corner but also to inform ongoing research and development on paths taken in the IETF.

3. The RTG WG SHOULD play a role in the engagement with research and development since the 'Future of Internet Routing' (FIR) is at the heart of its charter ("The Routing Area working group (RTGWG) is chartered to provide a venue to discuss, evaluate, support and develop proposals for new work in the Routing Area" [RTGWGref]), a role that goes beyond the "specific small topics that do not fit with an existing working group" [RTGWGref].

Following on the cornerstones outlined above, we specifically suggest to the RTG WG, aligned with its charter to consider the following actions:

1. Establish suitable efforts within the RTG WG (e.g., as a sub-group) OR

2. Support the establishment of suitable efforts as a standalone FIR WG (or special interest group) OR

3. Support the establishment of suitable efforts within the IRTF, where those efforts directly liaise with the RTG WG through regular updates in its meetings.

7. Security Considerations

Section 4.3 outlines a number of security issues that may occur outside the solution-specific security considerations, such as interactions between protocol behaviours that were previously untested as a combination. With that in mind, security considerations for a wider architectural approach to routing must have the security of the overall routing system as the main goal, not merely the security of a single solution.

8. Privacy Considerations

Protecting user privacy is very important. This extends beyond ensuring that user data cannot be examined in transit, and also requires that a process that inspects the network traffic should not be able to determine which applications or what types of application a user is running.

This makes it critically important to minimize or entirely avoid user and/or application information to be directly used for routing purposes. Instead, applications (or users) should express requirements for traffic delivery in a manner that does not reveal information about the user.

Encryption of user data, which is a common technique to protect user privacy, may obscure information that has previously been used to perform enhanced routing (such as by inspecting or hashing on payload fields), demonstrating that new requirements (here on privacy) may negatively impact previously accepted solutions.

9. IANA Considerations

This draft does not request any IANA action.

10. Acknowledgements

Many thanks go to Daniel King, Mohamed Boucadair for their comments to the text and to Lixia Zhang for raising important questions related to the possible architectural nature of the discussion needed.

11. Contributors

Adrian Farrel
Email: adrian@olddog.co.uk

12. Informative References

[CARDS] Khandaker, K., Trossen, D., Khalili, R., Despotovic, Z., Hecker, A., and G. Carle, "CArDS: Dealing a New Hand in Reducing Service Request Completion Times", Paper IFIP Networking 2022, 2022.

[CLNPref] "Protocol for providing the connectionless-mode network service: Protocol specification - Part 1", standard ISO/IEC 8473-1:1998, 1998, <https://www.iso.org/standard/30931.html>.

[CONTENTref] Choi, J., Han, J., and E. Cho, "A survey on content-oriented networking for efficient content delivery", Paper IEEE Communications Magazine, 49(3): 121-127, May 2011., 2011, <https://ieeexplore.ieee.org/iel5/35/5723785/05723809.pdf>.

[DONAref] Koponen, T., Chawla, M., Chun, B.-G., Ermolinskiy, A., Kim, K. H., Shenker, S., and I. Stoica, "A data-oriented (and beyond) network architecture", Paper Proceedings of the 2007 conference on Applications, technologies, architectures, and protocols for computer communications, August 2007, 2007, <https://dl.acm.org/doi/10.1145/1282380.1282402>.

[HICNref] Carofiglio, G., Muscariello, L., Auge, J., Papalini, M., Sardara, M., and A. Compagno, "Enabling ICN in the Internet Protocol: Analysis and Evaluation of the Hybrid-ICN Architecture", Paper Proceedings of the 6th ACM Conference on Information-Centric Networking, 2019., 2019, <https://www.researchgate.net/publication/336344520_Enabling_ICN_in_the_Internet_Protocol_Analysis_and_Evaluation_of_the_Hybrid-ICN_Architecture>.

[I-D.bonica-6man-ext-hdr-update] Bonica, R. and T. Jinmei, "Inserting, Processing And Deleting IPv6 Extension Headers", Work in Progress, Internet-Draft, draft-bonica-6man-ext-hdr-update-07, 24 February 2022, <https://www.ietf.org/archive/id/draft-bonica-6man-ext-hdr-update-07.txt>.

[I-D.bryant-arch-fwd-layer-ps] Bryant, S., Chunduri, U., Eckert, T., and A. Clemm, "Forwarding Layer Problem Statement", Work in Progress, Internet-Draft, draft-bryant-arch-fwd-layer-ps-04, 24 January 2022, <https://www.ietf.org/archive/id/draft-bryant-arch-fwd-layer-ps-04.txt>.

[I-D.chunduri-lsr-isis-preferred-path-routing] Chunduri, U., Li, R., White, R., Contreras, L. M., Tantsura, J., and Y. Qu, "Preferred Path Routing (PPR) in IS-IS", Work in Progress, Internet-Draft, draft-chunduri-lsr-isis-preferred-path-routing-07, 12 November 2021, <https://www.ietf.org/archive/id/draft-chunduri-lsr-isis-preferred-path-routing-07.txt>.

[I-D.farrel-irtf-introduction-to-semantic-routing] Farrel, A. and D. King, "An Introduction to Semantic Routing", Work in Progress, Internet-Draft, draft-farrel-irtf-introduction-to-semantic-routing-04, 25 April 2022, <https://www.ietf.org/archive/id/draft-farrel-irtf-introduction-to-semantic-routing-04.txt>.

[I-D.gont-v6ops-ipv6-addressing-considerations]
Gont, F. and G. Gont, "IPv6 Addressing Considerations", Work in Progress, Internet-Draft, draft-gont-v6ops-ipv6-addressing-considerations-02, 1 June 2022, <https://www.ietf.org/archive/id/draft-gont-v6ops-ipv6-addressing-considerations-02.txt>.

[I-D.ietf-lisp-mn]
Farinacci, D., Lewis, D., Meyer, D., and C. White, "LISP Mobile Node", Work in Progress, Internet-Draft, draft-ietf-lisp-mn-11, 30 January 2022, <https://www.ietf.org/archive/id/draft-ietf-lisp-mn-11.txt>.

[I-D.ietf-lisp-rfc6830bis]
Farinacci, D., Fuller, V., Meyer, D., Lewis, D., and A. Cabellos, "The Locator/ID Separation Protocol (LISP)", Work in Progress, Internet-Draft, draft-ietf-lisp-rfc6830bis-38, 7 May 2022, <https://www.ietf.org/archive/id/draft-ietf-lisp-rfc6830bis-38.txt>.

[I-D.ietf-lisp-rfc6833bis]
Farinacci, D., Maino, F., Fuller, V., and A. Cabellos, "Locator/ID Separation Protocol (LISP) Control-Plane", Work in Progress, Internet-Draft, draft-ietf-lisp-rfc6833bis-31, 2 May 2022, <https://www.ietf.org/archive/id/draft-ietf-lisp-rfc6833bis-31.txt>.

[I-D.ietf-teas-rfc3272bis]
Farrel, A., "Overview and Principles of Internet Traffic Engineering", Work in Progress, Internet-Draft, draft-ietf-teas-rfc3272bis-16, 24 March 2022, <https://www.ietf.org/archive/id/draft-ietf-teas-rfc3272bis-16.txt>.

[I-D.ietf-v6ops-ipv6-ehs-packet-drops]
Gont, F., Hilliard, N., Doering, G., Kumari, W., Huston, G., and W. (. Liu, "Operational Implications of IPv6 Packets with Extension Headers", Work in Progress, Internet-Draft, draft-ietf-v6ops-ipv6-ehs-packet-drops-08, 11 June 2021, <https://www.ietf.org/archive/id/draft-ietf-v6ops-ipv6-ehs-packet-drops-08.txt>.

[I-D.irtf-panrg-path-properties]
Enghardt, T. and C. Krähenbühl, "A Vocabulary of Path Properties", Work in Progress, Internet-Draft, draft-irtf-

panrg-path-properties-05, 7 March 2022, <https://www.ietf.org/archive/id/draft-irtf-panrg-path-properties-05.txt>.

[I-D.irtf-panrg-what-not-to-do]
Dawkins, S., "Path Aware Networking: Obstacles to Deployment (A Bestiary of Roads Not Taken)", Work in Progress, Internet-Draft, draft-irtf-panrg-what-not-to-do-19, 26 March 2021, <https://www.ietf.org/archive/id/draft-irtf-panrg-what-not-to-do-19.txt>.

[I-D.jia-intarea-internet-addressing-gap-analysis]
Jia, Y., Trossen, D., Iannone, L., Mendes, P., Shenoy, N., Toutain, L., Chen, A. Y., and D. Farinacci, "Gap Analysis in Internet Addressing", Work in Progress, Internet-Draft, draft-jia-intarea-internet-addressing-gap-analysis-02, 6 March 2022, <https://www.ietf.org/archive/id/draft-jia-intarea-internet-addressing-gap-analysis-02.txt>.

[I-D.king-irtf-semantic-routing-survey]
King, D. and A. Farrel, "A Survey of Semantic Internet Routing Techniques", Work in Progress, Internet-Draft, draft-king-irtf-semantic-routing-survey-04, 30 May 2022, <https://www.ietf.org/archive/id/draft-king-irtf-semantic-routing-survey-04.txt>.

[I-D.li-apn-framework]
Li, Z., Peng, S., Voyer, D., Li, C., Liu, P., Cao, C., and G. Mishra, "Application-aware Networking (APN) Framework", Work in Progress, Internet-Draft, draft-li-apn-framework-05, 7 March 2022, <https://www.ietf.org/archive/id/draft-li-apn-framework-05.txt>.

[I-D.li-dyncast-architecture]
Li, Y., Iannone, L., Trossen, D., Liu, P., and C. Li, "Dynamic-Anycast Architecture", Work in Progress, Internet-Draft, draft-li-dyncast-architecture-03, 7 March 2022, <https://www.ietf.org/archive/id/draft-li-dyncast-architecture-03.txt>.

[I-D.liu-dyncast-ps-usecases]
Liu, P., Eardley, P., Trossen, D., Boucadair, M., Contreras, L. M., and C. Li, "Dynamic-Anycast (Dyncast) Use Cases and Problem Statement", Work in Progress, Internet-Draft, draft-liu-dyncast-ps-usecases-03, 7 March 2022, <https://www.ietf.org/archive/id/draft-liu-dyncast-ps-usecases-03.txt>.

[I-D.muscariello-intarea-hicn]
Muscariello, L., Carofiglio, G., Augé, J., Papalini, M., and M. Sardara, "Hybrid Information-Centric Networking", Work in Progress, Internet-Draft, draft-muscariello-intarea-hicn-04, 20 May 2020, <https://www.ietf.org/archive/id/draft-muscariello-intarea-hicn-04.txt>.

[I-D.trossen-bier-frrm]
Trossen, D., "Realizing Forward Requests Return Multicast Semantic with BIER", Work in Progress, Internet-Draft, draft-trossen-bier-frrm-00, 21 February 2022, <https://www.ietf.org/archive/id/draft-trossen-bier-frrm-00.txt>.

[I-D.trossen-icnrg-internet-icn-5glan]
Trossen, D., Robitzsch, S., Reed, M., Al-Naday, M., and J. Riihijarvi, "Internet Services over ICN in 5G LAN Environments", Work in Progress, Internet-Draft, draft-trossen-icnrg-internet-icn-5glan-04, 1 October 2020, <https://www.ietf.org/archive/id/draft-trossen-icnrg-internet-icn-5glan-04.txt>.

[ICNref] Barbera, D., Xylomenos, G., Ververidis, C., Siris, V., and N. Fotiou, "A Survey of information-centric networking research", Paper IEEE Communications Surveys and Tutorials, vol. 16, Iss. 2, Q2 2014, 2014, <https://www.scopus.com/record/display.uri?eid=2-s2.0-84901242669>.

[ILNP_PRIVACY]
Bhatti, S., Haywood, G., and R. Yanagida, "End-to-end privacy for identity and location with IP", Paper 2nd Workshop on New Internetworking Protocols, Architecture and Algorithms, 29th IEEE International Conference on Network Protocols, 2021.

[ISISref] "Intermediate System to Intermediate System intra-domain routeing information exchange protocol for use in conjunction with the protocol for providing the connectionless-mode network service", standard ISO/IEC 10589, 2002, <https://standards.iso.org/ittf/PubliclyAvailableStandards/c030932_ISO_IEC_10589_2002(E).zip>.

[ITUNET2030ref]
"Network 2030 Architecture Framework", Technical Specification ITU-T Focus Group on Technologies for

Network 2030, 2020, <https://www.itu.int/en/ITU-T/focusgroups/net2030/Documents/Network_2030_Architecture-framework.pdf>.

[MBONEref] Savetz, K., Randall, N., and Y. Lepage, "MBONE: Multicasting Tomorrow's Internet", Book IDG, 1996, <https://www.savetz.com/mbone/>.

[NDNref] Zhang, L., Afanasyev, A., and J. Burke, "Named Data Networking", Paper ACM SIGCOMM Computer Communication, Review 44(3): 66-73, 2014, 2014.

[PANRGref] "Path Aware Networking Research Group", RG Path Aware Networking Research Group, <https://datatracker.ietf.org/rg/panrg/about>.

[RESEARCHFIAref] Pan, J., Paul, S., and R. Jain, "A Survey of the Research on Future Internet Architectures", Paper IEEE Communications Magazine, vol. 49, no. 7, July 2011, 2014, <https://ieeexplore.ieee.org/document/5936152>.

[RFC1069] Callon, R. and H. Braun, "Guidelines for the use of Internet-IP addresses in the ISO Connectionless-Mode Network Protocol", RFC 1069, DOI 10.17487/RFC1069, February 1989, <https://www.rfc-editor.org/info/rfc1069>.

[RFC1104] Braun, H., "Models of policy based routing", RFC 1104, DOI 10.17487/RFC1104, June 1989, <https://www.rfc-editor.org/info/rfc1104>.

[RFC1112] Deering, S., "Host extensions for IP multicasting", STD 5, RFC 1112, DOI 10.17487/RFC1112, August 1989, <https://www.rfc-editor.org/info/rfc1112>.

[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>.

[RFC1479] Steenstrup, M., "Inter-Domain Policy Routing Protocol Specification: Version 1", RFC 1479, DOI 10.17487/RFC1479, July 1993, <https://www.rfc-editor.org/info/rfc1479>.

[RFC2210] Wroclawski, J., "The Use of RSVP with IETF Integrated Services", RFC 2210, DOI 10.17487/RFC2210, September 1997, <https://www.rfc-editor.org/info/rfc2210>.

[RFC2474] Nichols, K., Blake, S., Baker, F., and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, DOI 10.17487/RFC2474, December 1998, <https://www.rfc-editor.org/info/rfc2474>.

[RFC2730] Hanna, S., Patel, B., and M. Shah, "Multicast Address Dynamic Client Allocation Protocol (MADCAP)", RFC 2730, DOI 10.17487/RFC2730, December 1999, <https://www.rfc-editor.org/info/rfc2730>.

[RFC2776] Handley, M., Thaler, D., and R. Kermode, "Multicast-Scope Zone Announcement Protocol (MZAP)", RFC 2776, DOI 10.17487/RFC2776, February 2000, <https://www.rfc-editor.org/info/rfc2776>.

[RFC2909] Radoslavov, P., Estrin, D., Govindan, R., Handley, M., Kumar, S., and D. Thaler, "The Multicast Address-Set Claim (MASC) Protocol", RFC 2909, DOI 10.17487/RFC2909, September 2000, <https://www.rfc-editor.org/info/rfc2909>.

[RFC2992] Hopps, C., "Analysis of an Equal-Cost Multi-Path Algorithm", RFC 2992, DOI 10.17487/RFC2992, November 2000, <https://www.rfc-editor.org/info/rfc2992>.

[RFC3376] Cain, B., Deering, S., Kouvelas, I., Fenner, B., and A. Thyagarajan, "Internet Group Management Protocol, Version 3", RFC 3376, DOI 10.17487/RFC3376, October 2002, <https://www.rfc-editor.org/info/rfc3376>.

[RFC3618] Fenner, B., Ed. and D. Meyer, Ed., "Multicast Source Discovery Protocol (MSDP)", RFC 3618, DOI 10.17487/RFC3618, October 2003, <https://www.rfc-editor.org/info/rfc3618>.

[RFC3972] Aura, T., "Cryptographically Generated Addresses (CGA)", RFC 3972, DOI 10.17487/RFC3972, March 2005, <https://www.rfc-editor.org/info/rfc3972>.

[RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, DOI 10.17487/RFC4291, February 2006, <https://www.rfc-editor.org/info/rfc4291>.

[RFC4423] Moskowitz, R. and P. Nikander, "Host Identity Protocol (HIP) Architecture", RFC 4423, DOI 10.17487/RFC4423, May 2006, <https://www.rfc-editor.org/info/rfc4423>.

[RFC4581] Bagnulo, M. and J. Arkko, "Cryptographically Generated Addresses (CGA) Extension Field Format", RFC 4581, DOI 10.17487/RFC4581, October 2006, <https://www.rfc-editor.org/info/rfc4581>.

[RFC4607] Holbrook, H. and B. Cain, "Source-Specific Multicast for IP", RFC 4607, DOI 10.17487/RFC4607, August 2006, <https://www.rfc-editor.org/info/rfc4607>.

[RFC4655] Farrel, A., Vasseur, J.-P., and J. Ash, "A Path Computation Element (PCE)-Based Architecture", RFC 4655, DOI 10.17487/RFC4655, August 2006, <https://www.rfc-editor.org/info/rfc4655>.

[RFC4786] Abley, J. and K. Lindqvist, "Operation of Anycast Services", BCP 126, RFC 4786, DOI 10.17487/RFC4786, December 2006, <https://www.rfc-editor.org/info/rfc4786>.

[RFC5305] Li, T. and H. Smit, "IS-IS Extensions for Traffic Engineering", RFC 5305, DOI 10.17487/RFC5305, October 2008, <https://www.rfc-editor.org/info/rfc5305>.

[RFC5440] Vasseur, JP., Ed. and JL. Le Roux, Ed., "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, DOI 10.17487/RFC5440, March 2009, <https://www.rfc-editor.org/info/rfc5440>.

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

[RFC6308] Savola, P., "Overview of the Internet Multicast Addressing Architecture", RFC 6308, DOI 10.17487/RFC6308, June 2011, <https://www.rfc-editor.org/info/rfc6308>.

[RFC6740] Atkinson, RJ. and SN. Bhatti, "Identifier-Locator Network Protocol (ILNP) Architectural Description", RFC 6740, DOI 10.17487/RFC6740, November 2012, <https://www.rfc-editor.org/info/rfc6740>.

[RFC6830] Farinacci, D., Fuller, V., Meyer, D., and D. Lewis, "The Locator/ID Separation Protocol (LISP)", RFC 6830, DOI 10.17487/RFC6830, January 2013, <https://www.rfc-editor.org/info/rfc6830>.

[RFC7094] McPherson, D., Oran, D., Thaler, D., and E. Osterweil, "Architectural Considerations of IP Anycast", RFC 7094, DOI 10.17487/RFC7094, January 2014, <https://www.rfc-editor.org/info/rfc7094>.

[RFC7285] Alimi, R., Ed., Penno, R., Ed., Yang, Y., Ed., Kiesel, S., Previdi, S., Roome, W., Shalunov, S., and R. Woundy, "Application-Layer Traffic Optimization (ALTO) Protocol", RFC 7285, DOI 10.17487/RFC7285, September 2014, <https://www.rfc-editor.org/info/rfc7285>.

[RFC7348] Mahalingam, M., Dutt, D., Duda, K., Agarwal, P., Kreeger, L., Sridhar, T., Bursell, M., and C. Wright, "Virtual eXtensible Local Area Network (VXLAN): A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks", RFC 7348, DOI 10.17487/RFC7348, August 2014, <https://www.rfc-editor.org/info/rfc7348>.

[RFC7401] Moskowitz, R., Ed., Heer, T., Jokela, P., and T. Henderson, "Host Identity Protocol Version 2 (HIPv2)", RFC 7401, DOI 10.17487/RFC7401, April 2015, <https://www.rfc-editor.org/info/rfc7401>.

[RFC7450] Bumgardner, G., "Automatic Multicast Tunneling", RFC 7450, DOI 10.17487/RFC7450, February 2015, <https://www.rfc-editor.org/info/rfc7450>.

[RFC7665] Halpern, J., Ed. and C. Pignataro, Ed., "Service Function Chaining (SFC) Architecture", RFC 7665, DOI 10.17487/RFC7665, October 2015, <https://www.rfc-editor.org/info/rfc7665>.

[RFC7872] Gont, F., Linkova, J., Chown, T., and W. Liu, "Observations on the Dropping of Packets with IPv6 Extension Headers in the Real World", RFC 7872, DOI 10.17487/RFC7872, June 2016, <https://www.rfc-editor.org/info/rfc7872>.

[RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <https://www.rfc-editor.org/info/rfc8200>.

[RFC8231] Crabbe, E., Minei, I., Medved, J., and R. Varga, "Path Computation Element Communication Protocol (PCEP) Extensions for Stateful PCE", RFC 8231, DOI 10.17487/RFC8231, September 2017, <https://www.rfc-editor.org/info/rfc8231>.

[RFC8279] Wijnands, IJ., Ed., Rosen, E., Ed., Dolganow, A., Przygienda, T., and S. Aldrin, "Multicast Using Bit Index Explicit Replication (BIER)", RFC 8279, DOI 10.17487/RFC8279, November 2017, <https://www.rfc-editor.org/info/rfc8279>.

[RFC8296] Wijnands, IJ., Ed., Rosen, E., Ed., Dolganow, A., Tantsura, J., Aldrin, S., and I. Meilik, "Encapsulation for Bit Index Explicit Replication (BIER) in MPLS and Non-MPLS Networks", RFC 8296, DOI 10.17487/RFC8296, January 2018, <https://www.rfc-editor.org/info/rfc8296>.

[RFC8300] Quinn, P., Ed., Elzur, U., Ed., and C. Pignataro, Ed., "Network Service Header (NSH)", RFC 8300, DOI 10.17487/RFC8300, January 2018, <https://www.rfc-editor.org/info/rfc8300>.

[RFC8402] Filsfils, C., Ed., Previdi, S., Ed., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", RFC 8402, DOI 10.17487/RFC8402, July 2018, <https://www.rfc-editor.org/info/rfc8402>.

[RFC8596] Malis, A., Bryant, S., Halpern, J., and W. Henderickx, "MPLS Transport Encapsulation for the Service Function Chaining (SFC) Network Service Header (NSH)", RFC 8596, DOI 10.17487/RFC8596, June 2019, <https://www.rfc-editor.org/info/rfc8596>.

[RFC8677] Trossen, D., Purkayastha, D., and A. Rahman, "Name-Based Service Function Forwarder (nSFF) Component within a Service Function Chaining (SFC) Framework", RFC 8677, DOI 10.17487/RFC8677, November 2019, <https://www.rfc-editor.org/info/rfc8677>.

[RFC8684] Ford, A., Raiciu, C., Handley, M., Bonaventure, O., and C. Paasch, "TCP Extensions for Multipath Operation with Multiple Addresses", RFC 8684, DOI 10.17487/RFC8684, March 2020, <https://www.rfc-editor.org/info/rfc8684>.

[RFC8736] Venaas, S. and A. Retana, "PIM Message Type Space Extension and Reserved Bits", RFC 8736, DOI 10.17487/RFC8736, February 2020, <https://www.rfc-editor.org/info/rfc8736>.

[RFC8754] Filsfils, C., Ed., Dukes, D., Ed., Previdi, S., Leddy, J., Matsushima, S., and D. Voyer, "IPv6 Segment Routing Header (SRH)", RFC 8754, DOI 10.17487/RFC8754, March 2020, <https://www.rfc-editor.org/info/rfc8754>.

[RFC8763] Rahman, A., Trossen, D., Kutscher, D., and R. Ravindran, "Deployment Considerations for Information-Centric Networking (ICN)", RFC 8763, DOI 10.17487/RFC8763, April 2020, <https://www.rfc-editor.org/info/rfc8763>.

[RFC8799] Carpenter, B. and B. Liu, "Limited Domains and Internet Protocols", RFC 8799, DOI 10.17487/RFC8799, July 2020, <https://www.rfc-editor.org/info/rfc8799>.

[RFC8926] Gross, J., Ed., Ganga, I., Ed., and T. Sridhar, Ed., "Geneve: Generic Network Virtualization Encapsulation", RFC 8926, DOI 10.17487/RFC8926, November 2020, <https://www.rfc-editor.org/info/rfc8926>.

[RFC8986] Filsfils, C., Ed., Camarillo, P., Ed., Leddy, J., Voyer, D., Matsushima, S., and Z. Li, "Segment Routing over IPv6 (SRv6) Network Programming", RFC 8986, DOI 10.17487/RFC8986, February 2021, <https://www.rfc-editor.org/info/rfc8986>.

[RINAref] Day, J., "Patterns in Network Architecture: A Return to Fundamentals", Book Prentice Hall, 2008.

[RTGWGinterim] King (ed.), D., "Minutes interim-2022-rtgwg-01: Tue 08:00", Minutes Minutes RTG WG interim meeting on Semantic Routing at 21.06.2022, 2022, <https://datatracker.ietf.org/doc/minutes-interim-2022-rtgwg-01-202206210800/>.

[RTGWGref] "Routing Working Group Charter", WG Routing Working Group, <https://datatracker.ietf.org/wg/rtgwg/charter/>.

[SCIONref] Chuat, L., Legner, M., Basin, D., Hausherr, D., Hitz, S., Mueller, P., and A. Perrig, "The Complete Guide to SCION. From Design Principles to Formal Verification", Book Springer International Publishing AG, 2022., 2022, <https://link.springer.com/book/10.1007/978-3-031-05288-0>.

[SHIMv6ref] Naderi, H. and B. E. Carpenter, "Putting SHIM6 into practice", Paper 2014 Australasian Telecommunication Networks and Applications Conference (ATNAC), 2014, <https://dl.acm.org/doi/10.1145/1282380.1282402>.

[TOR] "The Tor Project", Website Tor Project, 2022, <https://www.torproject.org/>.

[weightedRef]
Lin, C-Y., Lo, J-H., and S-Y. Kuo, "Load-balanced anycast routing", Paper Proceedings of the Tenth International Conference on Parallel and Distributed Systems, 2004., 2004, <https://www.researchgate.net/publication/4083002_Load-balanced_anycast_routing>.

Authors' Addresses

Dirk Trossen
Huawei Technologies
Munich
Germany
Email: dirk.trossen@huawei.com

David Lou
Huawei Technologies
Munich
Germany
Email: zhe.lou@huawei.com

Sheng Jiang
Huawei Technologies
China
Email: jiangsheng@huawei.com