SRv6 Midpoint Protection
draft-chen-rtgwg-srv6-midpoint-protection-08

Abstract

The current local repair mechanism, e.g., TI-LFA, allows local repair actions on the direct neighbors of the failed node or link to temporarily route traffic to the destination. This mechanism could not work properly when the failure happens in the destination point. In SRv6 TE, the IPv6 destination address in the outer IPv6 header could be the segment endpoint of the TE path rather than the destination of the TE path. When the SRv6 endpoint fails, local repair couldn’t work on the direct neighbor of the failed endpoint either. This document defines midpoint protection for SRv6 TE path, which enables other nodes on the network to perform endpoint behaviors instead of the faulty node, update the IPv6 destination address to the other endpoint, and choose the next hop based on the new destination address.

Requirements Language

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

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.
Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

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

This Internet-Draft will expire on 12 January 2023.

Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Introduction .................................................. 3
2. SRv6 Midpoint Protection Mechanism .......................... 3
3. SRv6 Midpoint Protection Example ............................ 3
4. SRv6 Midpoint Protection Behavior ............................ 5
   4.1. Endpoint Node as Repair Node ............................ 5
   4.2. Endpoint x Node as Repair Node .......................... 6
5. Determining whether the Endpoint could Be Bypassed .......... 6
6. Security Considerations ....................................... 6
7. IANA Considerations .......................................... 7
8. Acknowledgements ............................................. 7
9. References .................................................. 7
   9.1. Normative References .................................... 7
   9.2. Informative References .................................. 8
Authors’ Addresses .............................................. 8
1. Introduction

The current mechanism, e.g., TI-LFA ([I-D.ietf-rtgwg-segment-routing-ti-lfa]), allows local repair actions on the direct neighbors of the failed node or link to temporarily route traffic to the destination. This mechanism could not work properly when the failure happens in the destination point. In SRv6 TE, the IPv6 destination address in the outer IPv6 header could be the segment endpoint of the TE path rather than the destination of the TE path ([RFC8986]). When the endpoint fails, local repair couldn’t work on the direct neighbor of the failed endpoint either. This document defines midpoint protection for SRv6 TE path, which enables other nodes on the network to perform endpoint behaviors instead of the faulty node. Update the IPv6 destination address to the other endpoint, and choose the next hop based on the new destination address.

2. SRv6 Midpoint Protection Mechanism

When an endpoint node fails, the packet needs to bypass the failed endpoint node and be forwarded to the next endpoint node of the failed endpoint. Only endpoint node can process SRH, Therefore, only endpoint nodes can perform midpoint protection. There are two stages or time periods after an endpoint node fails. The first is the time period from the failure until the IGP converges on the failure. The second is the time period after the IGP converges on the failure.

During the first time period, the packet will be sent to the direct neighbor of the failed endpoint node. After detecting the failure of its interface to the failed endpoint node, the neighbor forwards the packets around the failed endpoint node. It changes the IPv6 destination address with the IPv6 address of the next endpoint node (or the last or other reasonable endpoint node) which could avoid going through the failed endpoint.

During the second time period. There is no route to the failed endpoint node after the IGP converges. When a previous hop node of the failed endpoint node finds out that there is no route to the IPv6 destination address (of the failed endpoint node), it changes the IPv6 destination address with the IPv6 address of the next endpoint node. Note that the previous hop node may not be the direct neighbor of the failed endpoint node.

3. SRv6 Midpoint Protection Example

The topology in Figure 1 illustrates an example of network topology with SRv6 enabled on each node.
In this document, an end SID at node n with locator block B is represented as B:n. An end.x SID at node n towards node k with locator block B is represented as B:n:k. A SID list is represented as \(<S_1, S_2, S_3>\) where \(S_1\) is the first SID to visit, \(S_2\) is the second SID to visit and \(S_3\) is the last SID to visit along the SRv6 TE path.

In the reference topology, suppose that Node N1 is an ingress node of SRv6 TE path going through N3 and N4. Node N1 steers a packet into a segment list \(<B:2, B:3, B:4>\).

When node N3 fails, the packet needs to bypass the failed endpoint node and be forwarded to the next endpoint node after the failed endpoint in the TE path. When outbound interface failure happens in the Repair Node (which is not limited to the previous hop node of the failed endpoint node), it performs the proxy forwarding as follows:

During the first time period (i.e., before the IGP converges), node N2 (direct neighbor of N3) as a Repair Node forwards the packets around the failed endpoint N3 after detecting the failure of the outbound interface to the endpoint B:3. It changes the IPv6 destination address with the next sid B:4. N2 detects the failure of outbound interface to B:4 in the current route, it could use the normal Ti-LFA repair path to forward the packet, because it is not directly connected to the node N4. N2 encapsulates the packet with the segment list \(<B:5:6>\) as a repair path.

During the second time period (i.e., after the IGP converges), node N2 does not have any route to the failed endpoint N3 in its FIB. Node N2, as a Repair Node, forwards the packets around the failed endpoint N3 to the next endpoint node (e.g., N4) directly. There is no need to check whether the failed endpoint node is directly connected to N2. N2 changes the IPv6 destination address with the next sid B:4. Since IGP has completed convergence, it forwards packets directly based on the IGP SPF path.
4. SRv6 Midpoint Protection Behavior

A node N protecting the failure of an endpoint node on a SRv6 path may be one of the following types:

* a transit node: The transit node cannot process SRH. Therefore, only Ti-LFA can be executed on the transit node, but not midpoint protection.

* an endpoint node: the destination address (DA) of the packet received by N is a N’s local END SID.

* an endpoint x node (i.e., an endpoint with cross-connect node): the destination address (DA) of the packet received by N is a N’s local End.X SID with an array of layer 3 adjacencies.

This section describes the behavior of each of these nodes as a repair node for the two time periods after the endpoint node fails.

4.1. Endpoint Node as Repair Node

When the Repair Node is an endpoint node, it provides fast protections for the failure through executing the following procedure after looking up the FIB for the updated DA.

IF the primary outbound interface used to forward the packet failed
IF NH = SRH && SL != 0 and
the failed endpoint is directly connected to Repair Node THEN
SL decreases; update the IPv6 DA with SRH[SL];
FIB lookup on the updated DA;
forward the packet according to the matched entry;
ELSE
forward the packet according to the backup nexthop;
ELSE IF there is no FIB entry for forwarding the packet THEN
IF NH = SRH && SL != 0 THEN
SL decreases; update the IPv6 DA with SRH[SL];
FIB lookup on the updated DA;
forward the packet according to the matched entry;
ELSE
drop the packet;
ELSE
forward accordingly to the matched entry;
4.2. Endpoint x Node as Repair Node

When the Repair Node is an endpoint x node, it provides fast protections for the failure through executing the following procedure after updating DA.

IF the layer-3 adjacency interface is down THEN
   FIB lookup on the updated DA;
   IF the primary interface used to forward the packet failed THEN
      IF NH = SRH && SL != 0 and
         the failed endpoint directly connected to Repair Node THEN
         SL decreases; update the IPv6 DA with SRH[SL];
         FIB lookup on the updated DA;
         forward the packet according to the matched entry;
      ELSE
         forward the packet according to the backup nexthop;
   ELSE IF there is no FIB entry for forwarding the packet THEN
      IF NH = SRH && SL != 0 THEN
         SL decreases; update the IPv6 DA with SRH[SL];
         FIB lookup on the updated DA;
         forward the packet according to the matched entry;
      ELSE
         drop the packet;
   ELSE
      forward accordingly to the matched entry;

5. Determining whether the Endpoint could Be Bypassed

SRv6 Midpoint Protection provides a mechanism to bypass a failed endpoint. But in some scenarios, some important functions may be implemented in the bypassed failed endpoints that should not be bypassed, such as firewall functionality or In-situ Flow Information Telemetry of a specified path. Therefore, a mechanism is needed to indicate whether an endpoint can be bypassed or not. [I-D.li-rtgwg-enhanced-ti-lfa] provides method to determine whether enable SRv6 midpoint protection or not by defining a "no bypass" flag for the SIDs in IGP.

6. Security Considerations

This section reviews security considerations related to SRv6 Midpoint protection processing discussed in this document. To ensure that the Repair node does not modify the SRH header Encapsulated by nodes outside the SRv6 Domain. Only the segment within the SRH is same domain as the repair node. So it is necessary to check the skipped segment have same block as repair node.
7. IANA Considerations

This document makes no request of IANA.

Note to RFC Editor: this section may be removed on publication as an RFC.

8. Acknowledgements

9. References

9.1. Normative References

[I-D.ietf-lsr-isis-srv6-extensions]

[I-D.ietf-lsr-ospfv3-srv6-extensions]


9.2. Informative References

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

[I-D.ietf-rtgwg-segment-routing-ti-lfa]

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

[I-D.li-rtgwg-enhanced-ti-lfa]

[I-D.sivabalan-pce-binding-label-sid]


Authors’ Addresses

SRv6 Egress Protection in Multi-home scenario
draft-cheng-rtgwg-srv6-multihome-egress-protection-01

Status of this Memo

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

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

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

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt

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

This Internet-Draft will expire on January 9 2023.

Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents.
Internet-Draft  Multi-home SRv6 Egress Protection  July 2022

carefully, as they describe your rights and restrictions with
respect to this document. Code Components extracted from this
document must include Simplified BSD License text as described in
Section 4.e of the Trust Legal Provisions and are provided without
warranty as described in the Simplified BSD License.

Abstract

This document describes a SRv6 egress node protection mechanism in
multi-home scenarios.

Table of Contents

1. Introduction ................................................. 2
2. Terminology .................................................. 3
   3.1. B-flag in Segment Routing Header ...................... 3
   3.2. Procedure of Multi-home Egress Protection on SRv6 TE Path 3
      3.2.1. Procedure on the Ingress Endpoint .................. 4
      3.2.2. Procedure on the Penultimate Endpoint .......... 6
   3.3. Procedure of Multi-home Egress Protection on SRv6 BE Path 7
4. Multi-home SRv6 Egress Protection Example .................... 8
5. IANA Considerations ......................................... 10
6. Security Considerations ..................................... 10
7. References .................................................. 11
   7.1. Normative References ................................... 11
   7.2. Informative References ................................. 12
8. Acknowledgments ............................................. 12
Authors’ Addresses ............................................. 12

1. Introduction

The fast protection of a transit node of a Segment Routing (SR) path
or tunnel is described in [I-D.ietf-rtgwg-segment-routing-ti-lfa] and
[I-D.hu-spring-segment-routing-proxy-forwarding]. [RFC8400] specifies the fast protection of egress node(s) of an MPLS TE LSP
tunnel including P2P TE LSP tunnel and P2MP TE LSP tunnel in details.
However, these documents do not discuss the fast protection of the
egress node of a Segment Routing for IPv6 (SRv6) path or tunnel.

[I-D.ietf-rtgwg-srv6-egress-protection] proposes mirror protection
mechanism and presents protocol extensions for the fast protection
of the egress node of a SRv6 path or tunnel. However, the mechanism
provided in this document is relatively complex. It is necessary to
configure the Mirror SID for the protected egress node on the backup
egress node. The mirror relationship is distributed through IGP and
BGP protocols to automatically create mapping entries.
This document introduces a simplified protection mechanism of the egress node of a SRv6 path, and only covers the scenario of direct connection to the egress nodes. Only expanding the data plane can perform fast path switching in case of egress node failure.

2. Terminology

The following terminologies are used in this document.

SR: Segment Routing
SRv6: SR for IPv6
SRH: Segment Routing Header
SID: Segment Identifier
CE: Customer Edge
PE: Provider Edge
VPN: Virtual Private Network

3. Multi-home SRv6 Egress Protection Mechanism

This section describes the mechanism of SRv6 path egress protection in multi-home scenarios and the extension of SRH extension header.

3.1. B-flag in Segment Routing Header

[RFC8754] describes the Segment Routing Header (SRH) and how SR capable nodes use it. The SRH contains an 8-bit "Flags" field.

This document defines the following bit in the SRH Flags field to carry the B-flag:

```
0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+-+-
|               |B|
|               |
+-+-+-+-+-+-+-+-+-
```

Where:

- B-flag: The marking bit of carrying backup SID in segment list. If the B-flag is set to 1, a backup SID is carried in the segment list.

3.2. Procedure of Multi-home Egress Protection on SRv6 TE Path

The Figure 1 is used to explain the multi-home egress node protection mechanism.
3.2.1. Procedure on the Ingress Endpoint

In the multi-home or dual-home scenario, after the ingress node learns the multi-home or dual-home route through routing protocol, it determines the optimal path and suboptimal path according to the route optimization strategy. The egress node on the optimal path is an primary egress, and the SID of the primary egress node is used as the primary SID. The egress node on the suboptimal path is an backup egress, and the SID of the backup egress node is used as the backup SID.

On the path forwarded based on SRv6 TE policy, when the ingress node encapsulates the SRH extension header, judge whether the primary VPN SID of the egress node (PE1) has a backup SID. If yes, insert the backup SID into the position of SRH[Last Entry], and set B-flag to 1 to identify that the backup SID has been carried in the last.
position of the segment list, then the value of SL is set to n-1. The format of SRH extension header filling is shown in the following figure 2.

```
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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Next Header   |  Hdr Ext Len  | Routing Type  |    SL = n-1   |
|+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Last Entry=n  |Flags(B-flag=1)|              Tag              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|      Active SID (Segment List[0], 128 bits IPv6 address)      |
|                                                               |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                           ...                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|      Segment List[n-1] (128 bits IPv6 address)               |
|                                                               |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|      Backup SID (Segment List[n],128 bits IPv6 value)        |
|                                                               |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
//               Optional Type Length Value objects (variable)  //
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2

3.2.2. Procedure on the Penultimate Endpoint

Normally, the traffic is forwarded along the path P1->P2->PE3->CE2. When primary egress node (PE3) fails, P2 finds out that the PE3’s SID is unreachable and the B-flag value is set. Then P2 modifies the destination address of the packet to SRH[Last Entry] which is the backup SID, and sends the modified packet to backup egress node (PE4). Through this method P2 can provide fast protection for the egress failure.

The detailed processing can be described in two cases according to the endpoint behavior of the destination address of the packet received by P2.

The behavior of the local endpoint is END.X

When receiving a packet destined to a local End.X SID whose outgoing interface is down, the penultimate endpoint acting as a Repair Node can provide fast protection for the failure of directly connected egress nodes after SL decreasing through executing the following procedures.

IF B-flag = 1 THEN
   IF SL = 0 and the failed egress node is directly connected to Repair Node THEN
      Update the IPv6 DA with SRH[Last Entry];
      FIB lookup on the updated DA;
      Forward the packet according to the matched entry;
   ELSE IF SL = 1 and SRH[1] and SRH[0] are the SIDs of the failed egress node directly connected to Repair Node THEN
      Update the IPv6 DA with SRH[Last Entry];
      FIB lookup on the updated DA;
      Forward the packet according to the matched entry;

The behavior of the local endpoint is END

After looking up the FIB for the updated DA with Segment List[Segments Left] and SL decreasing, in the following two cases, the penultimate endpoint acting as a Repair Node can provide fast protections for the failure of directly connected egress nodes through executing the following procedure.
Case 1: For the packet whose Next Header is SRH and Segments Left is equal to 1, perform the following processing:

IF B-flag = 1 and SRH[1] and SRH[0] are the SIDs of the failed egress node directly connected to Repair Node THEN
Update the IPv6 DA with SRH[Last Entry];
FIB lookup on the updated DA;
Forward the packet according to the matched entry;

Case 2: For the packet whose Next Header is SRH and Segments Left is equal to 0, perform the following processing:

IF B-flag = 1 and the failed egress endpoint is directly connected to Repair Node THEN
Update the IPv6 DA with SRH[Last Entry];
FIB lookup on the updated DA;
Forward the packet according to the matched entry;

When the packet arrives at PE4, PE4 removes the outer IPv6 header, and forwards the exposed inner packet.

After the route convergence is completed, the ingress node (PE1) will reselect the forwarding path for the traffic to VPN, and switch the path P1->P3->P4->PE4->CE2 to the CE to the egress node (PE4). After that, P2 no longer needs to forward the packet with the destination address of PE3.

Considering that the egress node may check the consistency between the segment list and the destination address, for the packet with B-flag 1, as long as the destination address is the same as any one of SRH[0] or SRH[Last Entry], it is considered to be consistent.

In addition, when a penultimate endpoint using non-PSP-flavored SID receives a packet with B-flag of 1, it is recommended to directly remove the SRH extension header after replacing the destination address with SRH[Last Entry].

3.3. Procedure of Multi-home Egress Protection on SRv6 BE Path

The multi-home egress node protection processing on the SRv6 BE path is consistent with that on the SRv6 TE path, except that the ingress...
node is required to add an SRH extension header with the active SID, backup SID and B-flag when encapsulating the outer IPv6 packet header.

In the multi-home scenario egress node scenario, the ingress node determines the active SID (PE3’s SID) and the backup SID (PE4’s SID) of the egress node through the optimization strategy of the routing protocol.

When the traffic from PE1 to CE2 is forwarded through the SRv6 BE path, in order to realize the fast protection of egress node failure, when the ingress node adds an outer IPv6 packet header to the forwarded packet, it must encapsulate the SRH extension header at the same time. The contents filled in the SRH extension header are the same as Figure 2 in Section 3.2.1, in which the segment list only fills in the active SID and backup SID, the SL is set to 0, the last entry is set to 1, and the B-flag is set to 1. The active SID is used as the destination address of the outer IP packet header.

Normally, because the destination address of the packet is the active SID (PE3’s SID), P1 and P2 will forward the packet to PE3 according to the destination address.

Once PE3 fails, the processing of the penultimate endpoint is the same as that on the SRv6 TE path. When P2 finds out that the route to the directly connected egress node PE3 is unreachable, if the B-flag is 1, modify the destination address to the backup SID in SRH[1], and send the packet to the updated destination address.

4. Multi-home SRv6 Egress Protection Example

Figure 3 shows an example of protecting egress PE3 of a SRv6 TE path, which is from ingress PE1 to egress PE3.
In this document, a SID list is represented as <S1, S2, S3> where S1 is the first SID to visit, S2 is the second SID to visit and S3 is the last SID to visit along the SRv6 path.

In Figure 3, Both CE2 and CE3 are dual home to PE3 and PE4. PE1 has a locator A0:1::/64. P1 has a locator A1:1::/64. P2 has a locator A2:1::/64 and END.X SID A2:1::A100. PE3 has a locator A3:1::/64 and a VPN SID A3:1::B100. PE4 has a locator A4:1::/64 and VPN SID A4:1::B100. The traffic from CE1 to CE2 is forwarded along the path PE1->P1->P2->PE3. After the configuration, PE1 determines that PE3’s backup SID is PE4’s VPN SID through the routing optimization strategy of BGP.

In normal operations, after receiving a packet with destination PE3, P2 forwards the packet to PE3 according to its FIB. When PE3 receives the packet, it sends the packet to CE2.

When PE1 receives the packet from CE1 to CE2, PE1 encapsulates the packet with IPv6 header. The segment list in SRH is designed as <A0:1::1, A1:1::1, A2:1::A100, A3:1::B100, A4:1::B100>. The SL is set to 3, the Last Entry is set to 4, and B-flag is set to 1.

When P2 receives a packet destined to END.X SID A2:1::A100, in normal operations, it forwards the packet with source A0:1::1 and
destination PE3’s VPN SID A3:1::B100 from the link between P2 and PE3 according to END.X SID.

When PE3 fails, P2 receives the packet to be sent to PE3’s VPN SID A3:1::B100. P2 finds that the outgoing interface is down. If the B-flag is 1, P2 changes the destination address of the packet with the backup SID of SRH[4], removes SRH extension header and sends the modified packet to A4:1::B100.

When PE4 receives the modified packet, it decapsulates the packet and forwards the decapsulated packet by executing End.DT6 behavior for an End.DT6 SID instance.

5. IANA Considerations

This document requests that IANA allocate the following registration in the "Segment Routing Header Flags" sub-registry for the "Internet Protocol Version 6 (IPv6) Parameters" registry maintained by IANA:

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>B-flag</td>
<td>This document</td>
</tr>
</tbody>
</table>

6. Security Considerations

[RFC8754] defines the notion of an SR domain and use of SRH within the SR domain. The use of egress protection mechanism described in this document is restricted to an SR domain. For example, similar to the SID manipulation, B-flag manipulation is not considered as a threat within the SR domain. Procedures for securing an SR domain are defined in section 5.1 and section 7 of [RFC8754].

This document does not impose any additional security challenges to be considered beyond security threats described in [RFC8754], [RFC8679] and [RFC8986].
7. References

7.1. Normative References


7.2. Informative References

TBD

8. Acknowledgments

The authors would like to thank the following for their valuable contributions of this document:

Yisong Liu
China Mobile

Authors’ Addresses

Weiqiang Cheng
China Mobile
Email: chengweiqiang@chinamobile.com

Wenying Jiang
China Mobile
Email: jiangwenying@chinamobile.com

Changwang Lin
New H3C Technologies
Email: linchangwang.04414@h3c.com

Zhibo Hu
Huawei Technologies
Email: huzhibo@huawei.com

Yuanxiang Qiu
New H3C Technologies
Email: qiuyuanxiang@h3c.com
Networks Connecting to Hybrid Cloud DCs: Gap Analysis
draft-ietf-rtgwg-net2cloud-gap-analysis-09

Abstract

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

Status of this Memo

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

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

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

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt

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

This Internet-Draft will expire on November 15, 2022.
Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Introduction...................................................3
2. Conventions used in this document................................3
3. Gap Analysis for Accessing Cloud Resources.....................4
   3.1. Multiple PEs connecting to virtual CPEs in Cloud DCs........6
   3.2. Access Control for workloads in the Cloud DCs.............6
   3.3. NAT Traversal.............................................7
   3.4. BGP between PEs and remote CPEs via Internet..............7
   3.5. Multicast traffic from/to the remote edges................8
4. Gap Analysis of Traffic over Multiple Underlay Networks........9
5. Aggregating VPN paths and Internet paths........................10
   5.1. Control Plane for Cloud Access via Heterogeneous Networks11
   5.2. Using BGP UPDATE Messages................................12
   5.2.1. Lacking identifier for different traffic in Cloud DCs12
   5.2.2. Missing attributes in Tunnel-Encap...................12
   5.3. SECURE-EVPN/BGP-EDGE-DISCOVERY........................12
   5.4. SECURE-L3VPN.............................................13
   5.5. Preventing attacks from Internet-facing ports.............14
6. Gap Summary...................................................14
7. Manageability Considerations..................................15
8. Security Considerations.......................................16
9. IANA Considerations...........................................16
10. References....................................................16
   10.1. Normative References.................................16
   10.2. Informative References.................................16
11. Acknowledgments..............................................17
1. Introduction

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

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

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

2. Conventions used in this document

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

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

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

OnPrem: On Premises data centers and branch offices
3. Gap Analysis for Accessing Cloud Resources

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

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

The MEF’s Cloud Service Architecture [MEF-Cloud] describes many scenarios of enterprises connecting to cloud DC. Including network operators using Overlay paths over an LTE network or the public Internet for the last mile access where the VPN service providers cannot provide the required physical infrastructure. In some scenarios, some overlay edge nodes may not be directly attached to the PEs that participate to the delivery and the operation of the enterprise’s VPN.

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

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

![Diagram of VPN Extension to Cloud DC](image)

**Figure 1: VPN Extension to Cloud DC**

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

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

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

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

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

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

3.2. Access Control for workloads in the Cloud DCs

There is widespread diffusion of access policy for Cloud Resource, some of which is not easy for verification and validation. Because there are multiple parties involved in accessing Cloud Resources, policy enforcement points are not easily visible for policy refinement, monitoring, and testing.
The current state of the art for specifying access policies for Cloud Resources could be improved by having automated and reliable tools to map the user-friendly (natural language) rules into machine readable policies and to provide interfaces for enterprises to self-manage policy enforcement points for their own workloads.

3.3. NAT Traversal

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

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

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

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

3.4. BGP between PEs and remote CPEs via Internet

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

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

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

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

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

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

3.5. Multicast traffic from/to the remote edges

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

This problem can be solved by selecting one floating PE as the CPE’s Designated Forwarder, like TRILL’s Appointed Forwarders [RFC6325].
4. Gap Analysis of Traffic over Multiple Underlay Networks

The hybrid Cloud DCs are often interconnected by multiple types of underlay networks, such as VPN, the public Internet, wireless and wired infrastructures, etc. Sometimes the enterprises’ VPN providers do not have direct access to the Cloud DCs that host the enterprises’ applications or workloads.

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

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

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

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

- Interworking with the MPLS VPN control plane. An overlay edge can have some ports facing the MPLS VPN network over which packets can be forwarded without encryption and some ports facing the public Internet over which sensitive traffic needs to be encrypted.
- Scalability: NHRP/DSVPN/DMVPN work fine with small numbers of edge nodes. When a network has more than 100 nodes, these protocols do not scale well.
- NHRP does not have the IPsec attributes, which are needed for peers to build Security Associations over the public Internet.
- NHRP messages do not have any field to encode the C-PE supported encapsulation types, such as IPsec-GRE or IPsec-VxLAN.
- NHRP messages do not have any field to encode C-PE Location identifiers, such as Site Identifier, System ID, and/or Port ID.
- NHRP messages do not have any field to describe the gateway(s) to which the C-PE is attached. When a C-PE is instantiated in a Cloud DC, it is desirable for the C-PE’s owner to be informed about how and where the C-PE is attached.
- NHRP messages do not have any field to describe C-PE’s NAT properties if the C-PE is using private IPv4 addresses, such as the NAT type, Private address, Public address, Private port, Public port, etc.

5. Aggregating VPN paths and Internet paths

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

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

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

- vCPE in a cloud DCs needs to communicate with its controller of the properties of the directly connected underlay networks.

- Need Controller-facilitated IPsec SA attributes and NAT information distribution
  - The controller facilitates and manages the peer authentication for all IPsec tunnels terminated at the vCPEs.

- Establishing and managing the topology and reachability for services attached to the vCPEs in Cloud DCs.
  - This is for the overlay layer’s route distribution, so that a vCPE can populate its overlay routing table with...
entries that identify the next hop for reaching a specific route/service attached to the vCPEs.

5.2. Using BGP UPDATE Messages

5.2.1. Lack ways to differentiate traffic in Cloud DCs

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

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

5.2.2. Miss attributes in Tunnel-Encap

[RFC9012] describes the BGP UPDATE Tunnel Path Attribute that advertises endpoints’ tunnel encapsulation capabilities for the respective attached client routes encoded in the MP-NLRI Path Attribute. The receivers of the BGP UPDATE can use any of the supported encapsulations encoded in the Tunnel Path Attribute for the routes encoded in the MP-NLRI Path Attribute.

Here are some of the issues raised by using [RFC9012] to distribute the property of client routes be carried by mixed of hybrid networks:

- [RFC9012] doesn’t have encoding methods to advertise that a route can be carried by a mixture of IPsec tunnels and other already supported tunnels.
- The mechanism defined in [RFC9012] does not facilitate the exchange of IPsec SA-specific attributes.

5.3. SECURE-EVPN/BGP-EDGE-DISCOVERY

[SECURE-EVPN] describes a solution that utilize BGP as control plane for the Scenario #1 described in [BGP-SDWAN-Usage]. It relies upon a
BGP cluster design to facilitate the key and policy exchange among PE devices to create private pair-wise IPsec Security Associations. [Secure-EVPN] attaches all the IPsec SA information to the actual client routes.

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

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

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

5.4. SECURE-L3VPN

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

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

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

- The [SECURE-L3VPN] assumes that a CPE "registers" with the RR. However, it does not say how. It assumes that the remote CPEs are pre-configured with the IPsec SA manually. For overlay networks to connect Hybrid Cloud DCs, Zero Touch Provisioning is expected. Manual configuration is not an option.
- The [SECURE-L3VPN] assumes that C-PEs and RRs are connected via an IPsec tunnel. For management channel, TLS/DTLS is more economical than IPsec. The following assumption made by [SECURE-L3VPN] can be difficult to meet in the environment where zero touch provisioning is expected:
  A CPE must also be provisioned with whatever additional information is needed in order to set up an IPsec SA with each of the red RRs.

- IPsec requires periodic refreshment of the keys. The [SECURE-L3VPN] does not provide any information about how to synchronize the refreshment among multiple nodes.

- IPsec usually sends configuration parameters to two endpoints only and lets these endpoints negotiate the key. The [SECURE-L3VPN] assumes that the RR is responsible for creating/managing the key for all endpoints. When one endpoint is compromised, all other connections may be impacted.

5.5. Preventing attacks from Internet-facing ports

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

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

6. Gap Summary

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

- For Accessing Cloud Resources

  a) Traffic Path Management: when a remote vCPE can be reached by multiple PEs of one provider VPN network, it is not straightforward to designate which egress PE should be used
to reach the remote vCPE based on applications or performance.

b) NAT Traversal: There is no automatic way for an enterprise’s network controller to be informed of the NAT properties for its workloads in Cloud DCs.

c) There is no loop prevention for the multicast traffic to/from remote vCPE in Cloud DCs.

A feature like Appointed Forwarder specified by TRILL is needed to prevent multicast data frames from looping around.

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

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

  BGP UPDATE propagates client’s routes information, but doesn’t distinguish between underlay networks.

- Issues of aggregating traffic over private paths and Internet paths

  a) Control plane messages for different overlay segmentations needs to be differentiated. User traffic belonging to different segmentations need to be differentiated.

  b) BGP Tunnel Encap doesn’t have ways to indicate a route or prefix that can be carried by both IPsec tunnels and VPN tunnels

  c) Missing clear methods in preventing attacks from Internet-facing ports

7. Manageability Considerations

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

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

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

9. IANA Considerations

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

10. References

10.1. Normative References


10.2. Informative References


11. Acknowledgments

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

Internet-Draft          Net2Cloud Gap Analysis

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

Linda Dunbar
Futurewei
Email: ldunbar@futurewei.com

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

Christian Jacquenet
Orange
Rennes, 35000
France
Email: Christian.jacquenet@orange.com
Dynamic Networks to Hybrid Cloud DCs Problem Statement
draft-ietf-rtgwg-net2cloud-problem-statement-14

Abstract

This document describes the network-related problems enterprises face today when interconnecting their branch offices with dynamic workloads in third-party data centers (a.k.a. Cloud DCs). There can be many problems associated with connecting to or among Clouds; the Net2Cloud problem statements are mainly for enterprises who already have traditional MPLS services and are interested in leveraging those networks (instead of completely abandoning them). This document aims to describe the problems of continuing using the MPLS networks when connecting workloads in the Cloud, and to clarify additional work in the IETF Routing area. Other problems are out of the scope of this document.

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

Status of this Memo

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

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

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

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt

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

This Internet-Draft will expire on December 29, 2022.

Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Introduction...................................................3
   1.1. Key Characteristics of Cloud Services....................3
   1.2. Connecting to Cloud Services............................3
   1.3. Reaching App instances in the optimal Cloud DC locations..4
2. Definition of terms............................................5
3. High Level Issues of Connecting to Cloud DCs...................6
   3.1. More BGP errors triggered by large number of peers........6
   3.2. Network failures that may lead to massive routes changes..6
   3.3. 5G Edge Clouds............................................7
1. Introduction

1.1. Key Characteristics of Cloud Services:

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

1.2. Connecting to Cloud Services

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

Dunbar, et al.
user experience. Conversely, an enterprise can easily shutdown applications and workloads whenever end-users are in motion (thereby modifying the networking connection of subsequently relocated applications and workloads). In addition, enterprises may wish to take advantage of more and more business applications offered by cloud operators.

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

- to access all workloads in the desired cloud DCs:
  Many enterprises include cloud in their disaster recovery strategy, such as enforcing periodic backup policies within the cloud, or running backup applications in the Cloud.

- Global reachability from different geographical zones, thereby facilitating the proximity of applications as a function of the end users’ location, to improve latency.

- Elasticity: prompt connection to newly instantiated applications at Cloud DCs when usages increase and prompt release of connection after applications at locations being removed when demands change.

- Scalable policy management: apply the appropriate policies to the newly instantiated application instances at any Cloud DC location.

1.3. Reaching App instances in the optimal Cloud DC locations

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

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

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

2. Definition of terms

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

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

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

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

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

VPC: Virtual Private Cloud is a virtual network dedicated to one client account. It is logically isolated from other virtual networks in a Cloud DC. Each client can launch his/her desired resources, such as compute, storage, or network functions into his/her VPC. Most Cloud
operators’ VPCs only support private addresses, some support IPv4 only, others support IPv4/IPv6 dual stack.

3. High Level Issues of Connecting to Cloud DCs

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

3.1. More BGP errors triggered by large number of peers

Many network service providers have limited number of BGP peers and usually have prior negotiated peering policies with their BGP peers. Cloud GWs need to peer with many more parties, via private circuits or IPsec over public internet. Many of those peering parties may not be traditional network service providers. Their BGP configurations practices might not be consistent, and some are done by less experienced personnel.

All those can contribute to increased BGP peering errors, such as capability mismatch, BGP cease notification, unwanted route leaks, missing Keepalives, etc.

3.2. Network failures that may lead to massive routes changes

As described in RFC7938, Cloud DC BGP might not have an IGP to route around link/node failures within the ASes. Fiber-cut is not uncommon within Cloud DCs or between sites. Sometimes, an entire cloud data center goes dark caused by a variety of reasons, such as too many changes and updates at once, changes of outside of maintenance windows, cybersecurity threats attacks, cooling failures, insufficient backup power, etc. When those events happen, massive numbers of routes need to be changed.

The large number of routes switching over to another site can also cause overloading that triggers more failures.
In addition, the routes (IP addresses) in a Cloud DC cannot be aggregated nicely, triggering very large number of BGP UPDATE messages when a failure occurs.

It might be more effective to do mass reroute, similar to EVPN [RFC7432] defined mass withdraw mechanism to signal a large number of routes being changed to remote PE nodes as quickly as possible.

3.3. 5G Edge Clouds

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

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

3.4. Security Issues

There are many aspects of security issues in terms of networking to clouds:

- Service instances in Cloud DCs are connected to users (enterprises) via Public IP ports which are exposed to the following security risks:

  a) Potential DDoS attack to the ports facing the untrusted network (e.g., the public internet), which may propagate to the cloud edge resources. To mitigate such security risk, it is necessary for the ports facing internet to enable Anti-DDoS features.

  b) Potential risk of augmenting the attack surface with inter-Cloud DC connection by means of identity spoofing, man-in-the-middle, eavesdropping or DDoS attacks. One example of mitigating such attacks is using DTLS to authenticate and encrypt MPLS-in-UDP encapsulation (RFC 7510).
- Potential attacks from service instances within the cloud. For example, data breaches, compromised credentials, and broken authentication, hacked interfaces and APIs, account hijacking.

- Securing user identity management, authentication, and access control mechanisms is important. Developing appropriate security mechanisms (including tools to assess the robustness of the enforced security policies) can enhance the confidence needed by enterprises to fully take advantage of Cloud Services.

Many Cloud operators offer monitoring services for data stored in Clouds, such as AWS CloudTrail, Azure Monitor, and many third-party monitoring tools to improve visibility to data stored in Clouds. But there is still underline security concerns on illegitimate data and workloads access.

3.5. Authorization and Identity Management

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

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

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

IETF hasn’t yet developed comprehensive specification for Identity management and data models for Cloud Authorizations.
3.6. API abstraction

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

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

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

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

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

3.7. DNS for Cloud Resources

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

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

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

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

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

3.8. NAT for Cloud Services

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

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

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

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

3.9. Cloud Discovery

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

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

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

4. Interconnecting Enterprise Sites with Cloud DCs

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

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

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

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

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

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

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

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

```
+------------------------+
\|    ,---.         ,---. |
\|   (TN-1 )       ( TN-2) |
\|    '--+-'  +---+  '-+-' |
\|      +----|vR1|----+   |
\|           ++--+        |
\|            |         +-+----+
\|            |        /Internet\ For External
\|            | Gateway +----------------------
\|            \ / to reach via Internet
\|             ++++++

+------------------------+
\|    ,---.         ,---. |
\|   (TN-1 )       ( TN-2) |
\|    '--+-'  +---+  '-+-' |
\|      +----|vR2|----+   |
\|           ++--+        |
\|            |         +-+----+
\|            |        / virtual\ For IPsec Tunnel
\|            | Gateway +----------------------
\|            \ / termination
\|             ++++++
\|             ++++++
\|             \ For Direct /customer\ 
\|             +------Gateway +--------+ +------
\|             \ / Connect \       / 
\|             ++++++     +--------+

+------------------------+
```

Figure 1: Examples of Multiple Cloud DC connections.
4.2. Inter-Cloud Interconnection

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

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

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

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

a) Utilize Cloud DC provided inter/intra-cloud connectivity services (e.g., AWS Transit Gateway) to connect workloads instantiated in multiple VPCs. Such services are provided with the cloud gateway to connect to external networks (e.g., AWS DirectConnect Gateway).

b) Hairpin all traffic through the customer gateway, meaning all workloads are directly connected to the customer gateway, so that communications among workloads within one Cloud DC must traverse through the customer gateway.

c) Establish direct tunnels among different VPCs (AWS’ Virtual Private Clouds) and VNET (Azure’s Virtual Networks) via client’s own virtual routers instantiated within Cloud DCs. DMVPN (Dynamic Multipoint Virtual Private Network) or DSVPN (Dynamic Smart VPN) techniques can be used to establish direct Multi-point-to-Point or multi-point-to multi-point tunnels among those client’s own virtual routers.
Approach a) usually does not work if Cloud DCs are owned and managed by different Cloud providers.

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

For the Approach c), DMVPN or DSVPN use NHRP (Next Hop Resolution Protocol) [RFC2735] so that spoke nodes can register their IP addresses & WAN ports with the hub node. The IETF ION (Internetworking over NBMA (non-broadcast multiple access) WG standardized NHRP for connection oriented NBMA network (such as ATM) network address resolution more than two decades ago.

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

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

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

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

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

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

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

Another roadblock is the lack of a standard way to express and enforce consistent security policies for workloads that not only use virtual addresses, but in which are also very likely hosted in different locations within the Cloud DC [RFC8192]. The current VPN path computation and bandwidth allocation schemes may not be flexible enough to address the need for enterprises to rapidly connect to dynamically instantiated (or removed) workloads and applications regardless of their location/nature (i.e., third party cloud DCs).
6. Problem with using IPsec tunnels to Cloud DCs
   As described in the previous section, many Cloud operators expose their gateways for external entities (which can be enterprises themselves) to directly establish IPsec tunnels. Enterprises can also instantiate virtual routers within Cloud DCs to connect to their on-premises devices via IPsec tunnels.

6.1. Scaling Issues with IPsec Tunnels

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

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

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

6.2. Poor performance when overlay public internet

   When large number of IPSec encap & decap are needed, the performance is degraded. NAT also adds performance burden.

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

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

7. End-to-End Security Concerns for Data Flows

Add description for Bucket 7 from Kausik
When IPsec tunnels established from enterprise on-premises CPEs are terminated at the Cloud DC gateway where the workloads or applications are hosted, some enterprises have concerns regarding traffic to/from their workload being exposed to others behind the data center gateway (e.g., exposed to other organizations that have workloads in the same data center).
To ensure that traffic to/from workloads is not exposed to unwanted entities, IPsec tunnels may go all the way to the workload (servers, or VMs) within the DC.

8. Requirements for Dynamic Cloud Data Center VPNs

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

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

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

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

10. IANA Considerations

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

11. References

11.1. Normative References

11.2. Informative References


12. Acknowledgments

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

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

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

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

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

Kausik Majumdar
Microsoft Azure
kmajumdar@microsoft.com
Abstract

This document describes a YANG model for configuration of Quality of Service (QoS) configuration in network devices. This document doesn’t describe QoS statistics counters.

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 8 January 2023.

Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.
1. Introduction

This document defines a base YANG [RFC6020] [RFC7950] model for Quality of Service (QoS) configuration parameters. QoS base modules define the basic building blocks to define a classifier, policy, action and target. The base models are augmented to include packet match fields and action parameters to define the Differentiated Services (DiffServ) module. Queues and schedulers are stitched as
part of diffserv policy model. Separate models have been defined for creating Queue policy and Scheduling policy. The DiffServ model is based on DiffServ architecture, and various references have been made to available standard architecture documents.

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

* 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

* 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.

* 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.

* WRED: Weighted Random Early Detection

2. QoS Model Design

A classifier consists of packets which may be grouped when a logical set of rules are applied on different packet header fields. The grouping may be based on different values or range of values of same packet header field, presence or absence of some values or range of values of a packet field or a combination thereof. The QoS classifier is defined in the ietf-qos-classifier module.

A classifier entry contains one or more packet conditioning functions. A packet conditioning function is typically based on direction of traffic and may drop, mark or delay network packets. A set of classifier entries with corresponding conditioning functions when arranged in order of priority represents a QoS policy. A QoS policy may contain one or more classifier entries. These are defined in ietf-traffic-policy module.
Actions are configured in line with respect to the policy module. These include marking, dropping or shaping. Actions are defined in the ietf-qos-action module.

A meter qualifies if the traffic arrival rate is based on agreed upon rate and variability. A meter is modeled based on commonly used algorithms in industry, Single Rate Tri Color Marking (srTCM) [RFC2697] meter, Two Rate Tri Color Marking (trTCM) [RFC2698] meter, and Single Rate Two Color Marking meter. Different vendors can extend it with other types of meters as well.

QoS operational model include QoS policy applied to an interface in each direction of traffic. For each QoS policy applied to an interface the model further includes counters for associated Classifiers, Meters and Queues in a particular direction. 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

DiffServ architecture [RFC3289] and [RFC2475] describe 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 ietf-diffserv module, this is realized by augmenting the QoS classification 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.
4. Modules Tree Structure

This document defines seven YANG modules - four QoS base modules, a
scheduler policy module, a queuing policy module and one DiffServ
module.

ietf-qos-classifier consists of classifier entries identified by a
classifier entry name. Each entry MAY contain a list of filter
entries. When no filter entry is present in a classifier entry, it
matches all traffic.

An ietf-traffic-policy module contains list of policy objects
identified by a policy name and policy type which MUST be provided.
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.

module: ietf-traffic-policy
   +--rw classifiers {classifier-template-feature}?
      |   +--rw classifier* [name]
      |      +--rw name                string
      |      +--rw description?        string
      |      +--rw filter-operation?   identityref
      |      +--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 1: ietf-traffic-policy tree diagram

ietf-qos-action module contains grouping of set of QoS actions.
These include metering, marking, dropping and shaping. Marking sets
DiffServ codepoint value in the classified packet. Color-aware and
Color-blind meters are augmented by vendor specific modules based on
the parameters defined in action module.

Choudhary, et al. Expires 8 January 2023
module: ietf-qos-action
  +--rw meters
    +--rw meter* [name]
      +--rw name              string
      +--rw (meter-type)?
      ...
module: ietf-diffserv

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

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

augment /policy:policies/policy:policy/policy:classifier
  /policy:action/policy:action-params:
+--:(dscp-marking)
  +--rw dscp
    +--rw dscp? inet:dscp
+--:(meter-inline)
  +--:(one-rate-two-color-meter-type)
    ...
  +--:(one-rate-tri-color-meter-type)
    ...
  +--:(two-rate-tri-color-meter-type)
    ...
+--:(meter-reference)
  +--rw meter
    +--rw name string
    +--rw type identityref
+--:(traffic-group-marking)
  +--rw traffic-group
    +--rw traffic-group? string
+--:(child-policy) {action:child-policy-feature}?  
  +--rw child-policy {child-policy-feature}?  
    +--rw policy-name? string
+--:(count) {action:count-feature}?  
  +--rw count {count-feature}?  
    +--rw count-action? empty
+--:(named-count) {action:named-counter-feature}?  
  +--rw named-counter {named-counter-feature}?  
    +--rw count-name-action? string
+--:(queue-inline)
  +--rw queue
    +--rw priority
    ...

Figure 3: ietf-diffserv tree diagram
module: ietf-queue-policy
  +--rw queue
    +--rw name?    string
    +--rw queue
      +--rw priority
      |     ...
      +--rw min-rate
      |     ...
      +--rw max-rate
      |     ...
      +--rw algorithmic-drop
      ...

augment /policy:policies/policy:policy/policy:classifier
  /policy:filter:
    +--rw (filter-params)?
    +--:(traffic-group-name)
      +--rw traffic-group
    ...

augment /policy:policies/policy:policy/policy:classifier
  /policy:action/policy:action-params:
    +--:(queue-template-name)
      +--rw queue-reference
      |     +--rw queue-name    string
      +--:(queue-inline)
        +--rw queue
          +--rw priority
          |     ...
          +--rw min-rate
          |     ...
          +--rw max-rate
          |     ...
          +--rw algorithmic-drop
          ...

Figure 4: ietf-queue-policy tree diagram
module: ietf-scheduler-policy

augment /policy:policies/policy:policy/policy:classifier
  /policy:filter:
    +--rw (filter-params)?
    |  +--:(filter-match-all)
    |     +--rw match-all-cfg
    ...

augment /policy:policies/policy:policy/policy:classifier
  /policy:action/policy:action-params:
    +--:(scheduler)
    |  +--rw scheduler
    |   |  +--rw min-rate
    |   |     ...
    |   |  +--rw max-rate
    |   |     ...
    |  +--:(queue-policy-name)
    |     +--rw queue-policy-name
    |     +--rw queue-policy    string

Figure 5: ietf-scheduler-policy 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-statistics* []
    |     |  ...
    |     +--ro named-statistics* []
    |     |  ...
    |     +--ro metering-statistics* []
    |     |  ...
    |     +--ro queueing-statistics* []
    |     |  ...

Figure 6: ietf-qos-oper tree diagram

5. Modules

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

5.1. ietf-traffic-policy
<CODE BEGINS> file "ietf-traffic-policy@2022-07-08.yang"
module ietf-traffic-policy {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-traffic-policy";
  prefix policy;

  import ietf-interfaces {
    prefix if;
  }
  import ietf-qos-action {
    prefix action;
  }
}

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 specification implementations."

Copyright (c) 2021 IETF Trust and the persons identified as 
authors of the code. All rights reserved.

Redistribution and use in source and binary forms, with or 
without modification, is permitted pursuant to, and subject 
to the license terms contained in, the 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 2022-07-08 {
  description
    "Initial version";
  reference
    "RFC XXXX: YANG Models for Quality of Service (QoS).";
}
feature policy-inline-classifier-config {
    description "This feature allows classifier configuration directly under policy.";
}

feature classifier-template-feature {
    description "This feature allows classifier as template configuration in a policy.";
}

identity policy-type {
    description "This base identity type defines policy-types";
}

identity diffserv {
    base policy-type;
    description "This defines ip policy-type";
}

identity ipv4-diffserv {
    base policy-type;
    description "This defines ipv4 policy-type";
}

identity ipv6-diffserv {
    base policy-type;
    description "This defines ipv6 policy-type";
}

identity filter-type {
    description "This is identity of base filter-type";
}

identity dscp {
    base filter-type;
    description "Differentiated services code point filter-type";
}

identity source-ipv4-prefix {
    base filter-type;
    description "source ipv4 prefix filter-type";
}

identity destination-ipv4-prefix {
    base filter-type;
    description "destination ipv4 prefix filter-type";
}


```yang

identity source-ipv6-prefix {
  base filter-type;
  description
    "source ipv6 prefix filter-type";
}

identity destination-ipv6-prefix {
  base filter-type;
  description
    "destination ipv6 prefix filter-type";
}

identity source-port {
  base filter-type;
  description
    "source port filter-type";
}

identity destination-port {
  base filter-type;
  description
    "destination port filter-type";
}

identity protocol {
  base filter-type;
  description
    "protocol type filter-type";
}

identity traffic-group-name {
  base filter-type;
  description
    "traffic-group filter type";
}

identity match-filter-operation {
  description
    "filter match logical operation type";
}

identity match-all-filter {
  base match-filter-operation;
  description
    "Classifier entry filter logical AND operation";
}

identity match-any-filter {
  base match-filter-operation;
  description
    "Classifier entry filter logical OR operation";
}

identity direction {
  description

```
"This is identity of traffic direction";
)
identity inbound {
    base direction;
    description
    "Direction of traffic coming into the network entry";
}
identity outbound {
    base direction;
    description
    "Direction of traffic going out of the network entry";
}

grouping filters {
    description
    "Filters types in a Classifier entry";
    leaf type {
        type identityref {
            base filter-type;
        }
        description
        "This leaf defines type of the filter";
    }
    leaf logical-not {
        type boolean;
        description
        "This is logical-not operator for a filter. When true, it indicates filter looks for absence of a pattern defined by the filter.";
    }
}

grouping generic-classifier-attr {
    description
    "Classifier generic attributes like name, description, operation type";
    leaf name {
        type string;
        description
        "classifier entry name";
    }
    leaf description {
        type string;
        description
        "classifier entry description statement";
    }
    leaf filter-operation {
        type identityref {

base match-filter-operation;
}
default "match-all-filter";
description
  "Filters are applicable as match-any or match-all filters";
}
}
grouping inline-attr {
description
  "attributes of inline classifier in a policy";
leaf inline {
  type boolean;
default "false";
description
  "Indication of inline classifier entry";
}
leaf filter-operation {
  type identityref {
    base match-filter-operation;
  }
default "match-all-filter";
description
  "Filters are applicable as match-any or match-all filters";
}
list filter{
  if-feature policy-inline-classifier-config;
must " ../inline = 'true' " {
    description
      "For inline filter configuration, inline attribute" +
      "must be true";
  }
  key "type logical-not";
  uses filters;
description
  "Filters configured inline in a policy";
}
}
grouping generic-policy-attr {
description
  "Policy Attributes";
leaf name {
  type string;
description
    "policy name";
}
leaf type {
  type identityref {
    base policy-type;
description "policy type";
}
leaf description {
  type string;
  description "policy description";
}

identity action-type {
  description "This base identity type defines action-types";
}
identity dscp-marking {
  base action-type;
  description "dscp marking action type";
}
identity meter-inline {
  base action-type;
  description "meter-inline action type";
}
identity meter-reference {
  base action-type;
  description "meter reference action type";
}
identity queue {
  base action-type;
  description "queue action type";
}
identity scheduler {
  base action-type;
  description "scheduler action type";
}
identity discard {
  base action-type;
  description "discard action type";
}
identity child-policy {
  if-feature action:child-policy-feature;
  base action-type;
  description
"child-policy action type";
}

identity count {
  if-feature action:count-feature;
  base action-type;
  description
    "count action type";
}

identity named-counter {
  if-feature action:named-counter-feature;
  base action-type;
  description
    "name counter action type";
}

grouping classifier-action-entry {
  description
    "List of Configuration of classifier & associated actions";
  list action {
    key "type";
    ordered-by user;
    description
      "Configuration of classifier & associated actions";
    leaf type {
      type identityref {
        base action-type;
      }
      description
        "This defines action type ";
    }
    choice action-params {
      description
        "Choice of action types";
    }
  }
}

container classifiers {
  if-feature classifier-template-feature;
  description
    "list of classifier entry";
  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;
container policies{
  description
    "list of policy templates";
  list policy{
    key "name type";
    description
      "policy template";
    uses generic-policy-attr;
    list classifier{
      key "name";
      ordered-by user;
      description
        "Classifier entry configuration in a policy";
      leaf name {
        type string;
        description
          "classifier entry name";
      }
      uses inline-attr;
      uses classifier-action-entry;
    }
  }
}

augment "/if:interfaces/if:interface" {
  description
    "Augments Diffserv Target Entry to Interface module";
  list qos-target-policy {
    key "direction type";
    description
      "policy target for inbound or outbound direction";
    leaf direction {
      type identityref {
        base direction;
      }
      description
        "Direction of the traffic flow either inbound or outbound";
    }
    leaf type {
      type identityref {
        base policy-type;
      }
      description
        "Policy entry type";
    }
}
5.2. ietf-qos-action

Figure 7: ietf-traffic-policy module
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 2022-07-08 {
  description
    "Initial version";
  reference
    "RFC XXXX: YANG Models for Quality of Service (QoS).";
}

feature child-policy-feature {
  description
    "This feature allows configuration of hierarchical policy.";
}

feature count-feature {
  description
    "This feature allows action configuration to enable
counter in a classifier";
}

feature named-counter-feature {
  description
    "This feature allows action configuration to enable
named counter in a classifier";
}

identity rate-unit-type {
  description
    "base rate-unit type";
}

identity bits-per-second {
  base rate-unit-type;
  description
    "bits per second identity";
}

identity kilo-bits-per-second {
  base rate-unit-type;
  description
    "kilo bits per second identity";
}

identity mega-bits-per-second {
  base rate-unit-type;
  description
    "mega bits per second identity";
}
} identity giga-bits-per-second {
    base rate-unit-type;
    description
        "mega bits per second identity";
}
identity percent {
    base rate-unit-type;
    description
        "percentage";
}
identity burst-unit-type {
    description
        "base burst-unit type";
}
identity bytes {
    base burst-unit-type;
    description
        "bytes";
}
identity kilo-bytes {
    base burst-unit-type;
    description
        "kilo bytes";
}
identity mega-bytes {
    base burst-unit-type;
    description
        "mega bytes";
}
identity millisecond {
    base burst-unit-type;
    description
        "milli seconds";
}
identity microsecond {
    base burst-unit-type;
    description
        "micro seconds";
}
identity red-threshold-unit {
    description
        "base red-unit type";
}
identity red-threshold-bytes {
    base red-threshold-unit;
    description
        "bytes";
} identity red-threshold-kb {
  base red-threshold-unit;
  description
  "kilo bytes";
}
identity red-threshold-mb {
  base red-threshold-unit;
  description
  "mega bytes";
}
identity red-threshold-ms {
  base red-threshold-unit;
  description
  "milli seconds";
}
identity red-threshold-us {
  base red-threshold-unit;
  description
  "micro seconds";
}
identity red-threshold-pc {
  base red-threshold-unit;
  description
  "per-centage";
}
identity red-threshold-pt {
  base red-threshold-unit;
  description
  "per-thousand";
}
identity red-threshold-pm {
  base red-threshold-unit;
  description
  "per-million";
}
identity wred-color-type {
  description
  "base wred color type";
}
identity wred-color-dscp {
  base wred-color-type;
  description
  "dscp wred color type";
}
identity probability-unit {
  description
  "base probability unit type";
identity probability-pc {
  base probability-unit;
  description
    "probability in percentage";
}

identity probability-pt {
  base probability-unit;
  description
    "probability in per thousand";
}

identity probability-pm {
  base probability-unit;
  description
    "probability in per million";
}

identity probability-denominator {
  base probability-unit;
  description
    "probability value is denominator value
     while numerator is always 1";
}

identity meter-type {
  description
    "This base identity type defines meter types";
}

identity one-rate-two-color-meter-type {
  base meter-type;
  description
    "one rate two color meter type";
}

identity one-rate-tri-color-meter-type {
  base meter-type;
  description
    "one rate three color meter type";
}

identity two-rate-tri-color-meter-type {
  base meter-type;
  description
    "two rate three color meter action type";
}

identity drop-type {
  description
    "drop algorithm";
}

identity tail-drop {
  base drop-type;
description
  "tail drop algorithm";
}
identity red {
  base drop-type;
  description
    "Random Early Detect drop algorithm";
}
identity wred {
  base drop-type;
  description
    "Weighted Random Early Detect drop algorithm";
}
identity conform-2color-meter-action-type {
  description
    "action type in a meter";
}
identity exceed-2color-meter-action-type {
  description
    "action type in a meter";
}
identity conform-3color-meter-action-type {
  description
    "action type in a meter";
}
identity exceed-3color-meter-action-type {
  description
    "action type in a meter";
}
identity violate-3color-meter-action-type {
  description
    "action type in a meter";
}
grouping rate-value-unit {
  leaf rate-value {
    type uint64;
    description
      "rate value";
  }
  leaf rate-unit {
    type identityref {
      base rate-unit-type;
    }
    description
      "rate unit";
  }
}
grouping burst {
    description  
      "burst value and unit configuration";
    leaf burst-value {
        type uint64;
        description  
          "burst value";
    }
    leaf burst-unit {
        type identityref {
            base burst-unit-type;
        }
        description  
          "burst unit";
    }
}

grouping threshold {
    description  
      "Threshold Parameters";
    container threshold {
        description  
          "threshold";
        choice threshold-type {
            case size {
                leaf threshold-size {
                    type uint64;
                    units "bytes";
                    description  
                      "Threshold size";
                } 
            } 
            case interval {
                leaf threshold-interval {
                    type uint64;
                    units "microsecond";
                    description  
                      "Threshold interval";
                } 
            } 
        } 
        description  
          "Choice of threshold type";
    } 
}
grouping drop {
    container drop {
        leaf drop-action {
            type empty;
            description
            "always drop algorithm";
        }
        description
        "the drop action";
    }
    description
    "always drop grouping";
}

grouping queuelimit {
    container qlimit-thresh {
        uses threshold;
        description
        "the queue limit";
    }
    description
    "the queue limit beyond which queue will not hold any packet";
}

grouping conform-2color-meter-action-params {
    description
    "meter action parameters";
    list conform-2color-meter-action-params {
        key "conform-2color-meter-action-type";
        ordered-by user;
        description
        "Configuration of basic-meter & associated actions";
        leaf conform-2color-meter-action-type {
            type identityref {
                base conform-2color-meter-action-type;
            }
            description
            "meter action type";
        }
        choice conform-2color-meter-action-val {
            description
            " meter action based on choice of meter action type";
        }
    }
}

grouping exceed-2color-meter-action-params {
    description
"meter action parameters";
list exceed-2color-meter-action-params {
  key "exceed-2color-meter-action-type";
  ordered-by user;
  description
    "Configuration of basic-meter & associated actions";
  leaf exceed-2color-meter-action-type {
    type identityref {
      base exceed-2color-meter-action-type;
    }
    description
      "meter action type";
  }
  choice exceed-2color-meter-action-val {
    description
      "meter action based on choice of meter action type";
  }
}
}
grouping conform-3color-meter-action-params {
  description
    "meter action parameters";
  list conform-3color-meter-action-params {
    key "conform-3color-meter-action-type";
    ordered-by user;
    description
      "Configuration of basic-meter & associated actions";
    leaf conform-3color-meter-action-type {
      type identityref {
        base conform-3color-meter-action-type;
      }
      description
        "meter action type";
    }
    choice conform-3color-meter-action-val {
      description
        "meter action based on choice of meter action type";
    }
  }
}

grouping exceed-3color-meter-action-params {
  description
    "meter action parameters";
  list exceed-3color-meter-action-params {
    key "exceed-3color-meter-action-type";

ordered-by user;
description
  "Configuration of basic-meter & associated actions";
leaf exceed-3color-meter-action-type {
  type identityref {
    base exceed-3color-meter-action-type;
  }
description
  "meter action type";
}
choice exceed-3color-meter-action-val {
  description
  " meter action based on choice of meter action type";
}
}
}

grouping violate-3color-meter-action-params {
  description
  "meter action parameters";
list violate-3color-meter-action-params {
  key "violate-3color-meter-action-type";
  ordered-by user;
  description
  "Configuration of basic-meter & associated actions";
  leaf violate-3color-meter-action-type {
    type identityref {
      base violate-3color-meter-action-type;
    }
    description
      "meter action type";
  }
  choice violate-3color-meter-action-val {
    description
      " meter action based on choice of meter action type";
  }
}
}

grouping one-rate-two-color-meter {
  container one-rate-two-color-meter {
    description
      "single rate two color marker meter";
    leaf committed-rate-value {
      type uint64;
      description
        "committed rate value";
    }
  }
}
leaf committed-rate-unit {
    type identityref {
        base rate-unit-type;
    }
    description
        "committed rate unit";
}
leaf committed-burst-value {
    type uint64;
    description
        "burst value";
}
leaf committed-burst-unit {
    type identityref {
        base burst-unit-type;
    }
    description
        "committed burst unit";
}
container conform-action {
    uses conform-2color-meter-action-params;
    description
        "conform action";
}
container exceed-action {
    uses exceed-2color-meter-action-params;
    description
        "exceed action";
}
}
description
    "single rate two color marker meter attributes";
}
grouping one-rate-tri-color-meter {
    container one-rate-tri-color-meter {
        description
            "single rate three color meter";
    }
    leaf committed-rate-value {
        type uint64;
        description
            "meter rate";
    }
    leaf committed-rate-unit {
        type identityref {
            base rate-unit-type;
        }
        description
            "committed rate unit";
    }
"committed rate unit";
}
leaf committed-burst-value {
  type uint64;
  description
    "committed burst size";
}
leaf committed-burst-unit {
  type identityref {
    base burst-unit-type;
  }
  description
    "committed burst unit";
}
leaf excess-burst-value {
  type uint64;
  description
    "excess burst size";
}
leaf excess-burst-unit {
  type identityref {
    base burst-unit-type;
  }
  description
    "excess burst unit";
}
container conform-action {
  uses conform-3color-meter-action-params;
  description
    "conform, or green action";
}
container exceed-action {
  uses exceed-3color-meter-action-params;
  description
    "exceed, or yellow action";
}
container violate-action {
  uses violate-3color-meter-action-params;
  description
    "violate, or red action";
}
}

description
  "one-rate-tri-color-meter attributes";
}

grouping two-rate-tri-color-meter {
  container two-rate-tri-color-meter {
}
description
  "two rate three color meter";
leaf committed-rate-value {
  type uint64;
  units "bits-per-second";
  description
    "committed rate";
}
leaf committed-rate-unit {
  type identityref {
    base rate-unit-type;
  }
  description
    "committed rate unit";
}
leaf committed-burst-value {
  type uint64;
  description
    "committed burst size";
}
leaf committed-burst-unit {
  type identityref {
    base burst-unit-type;
  }
  description
    "committed burst unit";
}
leaf peak-rate-value {
  type uint64;
  description
    "peak rate";
}
leaf peak-rate-unit {
  type identityref {
    base rate-unit-type;
  }
  description
    "committed rate unit";
}
leaf peak-burst-value {
  type uint64;
  description
    "committed burst size";
}
leaf peak-burst-unit {
  type identityref {
    base burst-unit-type;
  }
}
description
  "peak burst unit";
}
container conform-action {
  uses conform-3color-meter-action-params;
  description
    "conform, or green action";
}
container exceed-action {
  uses exceed-3color-meter-action-params;
  description
    "exceed, or yellow action";
}
container violate-action {
  uses violate-3color-meter-action-params;
  description
    "exceed, or red action";
}
}

description
  "two-rate-tri-color-meter attributes";
}
grouping meter {
  choice meter-type {
    case one-rate-two-color-meter-type {
      uses one-rate-two-color-meter;
      description
        "basic meter";
    }
    case one-rate-tri-color-meter-type {
      uses one-rate-tri-color-meter;
      description
        "one rate tri-color meter";
    }
    case two-rate-tri-color-meter-type {
      uses two-rate-tri-color-meter;
      description
        "two rate tri-color meter";
    }
    description
      "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
        "meter identifier";
    }
    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 meter-type;
            }
            mandatory true;
            description
            "This leaf defines type of the meter";
        }
        description
        "meter reference name";
    }
    description
    "meter reference";
}

grouping count {
    container count {
        if-feature count-feature;
        leaf count-action {
            type empty;
            description
            "count action";
        }
        description
        "the count action";
    }
}
grouping named-counter {
  container named-counter {
    if-feature named-counter-feature;
    leaf count-name-action {
      type string;
      description
      "count action";
    }
    description
    "the count action";
  }
  description
  "the count action grouping";
}

grouping discard {
  container discard {
    leaf discard {
      type empty;
      description
      "discard action";
    }
    description
    "discard action";
  }
  description
  "discard grouping";
}

grouping priority {
  container priority {
    leaf priority-level {
      type uint8;
      description
      "priority level";
    }
    description
    "priority attributes";
  }
  description
  "priority attributes grouping";
}

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

grouping dscp-marking {
    container dscp {
        leaf dscp {
            type inet:dscp;
            description
                "dscp marking";
        }
        description
            "dscp marking container";
    }
    description
        "dscp marking grouping";
}

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

grouping child-policy {
    container child-policy {
        if-feature child-policy-feature;
        leaf policy-name {
            type string;
            description
                "Hierarchical Policy";
        }
        description
            "Hierarchical Policy configuration container";
    }
    description
        "Grouping of Hierarchical Policy configuration";
}
grouping max-rate {
    container max-rate {
        uses rate-value-unit;
        uses burst;
        description
            "maximum rate attributes container";
    }
    description
        "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 red-threshold-unit;
        }
        description
            "unit of minimum red threshold";
    }
    leaf max-threshold-val {
        type uint64;
        description
            "maximum value of red threshold";
    }
    leaf max-threshold-unit {
        type identityref {
            base red-threshold-unit;
        }
        description
            "unit of maximum red threshold";
    }
    leaf weight {
        type uint8;
        description
            "the decay factor for the average queue size calculation. the numbers are 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 probability-unit;
}
description
  "probability unit type as defined
   by probability-unit";
} 
description
  "Random Early Detect 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
                "tail drop algorithm";
            }
            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 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 wred-color-type;
  }
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 configuration";
}
}
description
"Choice of Drop Algorithm";
}
description
"Algorithmic Drop configuration container";
}
description
"Queue configuration container";
}
description
"Queue grouping";
}
grouping scheduler {
    container scheduler {
        uses min-rate;
        uses max-rate;
        description
            "Scheduler configuration container";
        }
description
            "Scheduler configuration grouping";
    }
}<CODE ENDS>

Figure 8: ietf-qos-actions module

5.3. ietf-diffserv

<CODE BEGINS>
file "ietf-diffserv@2022-07-08.yang"
module ietf-diffserv {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-diffserv";
  prefix diffserv;

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

  organization
    "IETF Routing Area Working Group";
  contact
    "WG Web:  <https://datatracker.ietf.org/wg/rtgwg/>
    WG List:  <mailto:rtgwg@ietf.org>
    Editor:  Aseem Choudhary";

Choudhary, et al. Expires 8 January 2023 [Page 41]
description
"This module contains a collection of YANG definitions for configuring diffserv specification implementations.

Copyright (c) 2021 IETF Trust and the persons identified as authors of the code. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, is permitted pursuant to, and subject to the license terms contained in, the 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 2022-07-08 {
  description
  "Initial version";
  reference
  "RFC XXXX: A YANG Model for Quality of Service (QoS)";
}

identity meter-type {
  description
  "This base identity type defines meter types";
}

identity one-rate-two-color-meter-type {
  base meter-type;
  description
  "one rate two color meter type";
}

identity one-rate-tri-color-meter-type {
  base meter-type;
  description
  "one rate three color meter type";
}

identity two-rate-tri-color-meter-type {
  base meter-type;
  description
  "two rate three color meter action type";
}

grouping dscp {

list dscp {
  key "dscp-min dscp-max";
  description
    "list of dscp ranges";
  leaf dscp-min {
    type inet:dscp;
    description
      "Minimum value of dscp min-max range";
  }
  leaf dscp-max {
    type inet:dscp;
    must ". >= ../dscp-min" {
      error-message
        "The dscp-max must be greater than or equal to dscp-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 "source-ipv4-prefix";
    description
      "list of source ipv4 prefix";
    leaf source-ipv4-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 "destination-ipv4-prefix";
    description
      "list of destination ipv4 prefix";
    leaf destination-ipv4-prefix {
      type inet:ipv4-prefix;
      description
        "destination ipv4 prefix";
    }
  }
}

description
"Filter grouping containing list of destination ipv4 prefix";
}
grouping source-ipv6-prefix {
  list source-ipv6-prefix {
    key "source-ipv6-prefix";
    description
    "list of source ipv6 prefix";
    leaf source-ipv6-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 "destination-ipv6-prefix";
    description
    "list of destination ipv4 or ipv6 prefix";
    leaf destination-ipv6-prefix {
      type inet:ipv6-prefix;
      description
      "destination ipv6 prefix";
    }
  };
}
description
"Filter grouping containing list of destination ipv6 prefix";
}
grouping source-port {
  list source-port {
    key "source-port-min source-port-max";
    description
    "list of ranges of source port";
    leaf source-port-min {
      type inet:port-number;
      description
      "minimum value of source port range";
    }
    leaf source-port-max {
      type inet:port-number;
      must ". >= ../source-port-min" {
        error-message
        "The source-port-max must be greater than or equal to
        source-port-min";
      }
      description
      "maximum value of source port range";
    }
  };
}

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

grouping protocol {
  list protocol {
    key "protocol-min protocol-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 protocol-min {
      type uint8 {
        range "0..255";
      }
      description
    }
  }
  description
    "Filter grouping containing list of protocol ranges";
}
"minimum value of protocol range";
}
leaf protocol-max {
    type uint8 {
        range "0..255";
    }
    must ". >= ../protocol-min" {
        error-message
        "The protocol-max must be greater than or equal to
         protocol-min";
    }
    description
    "maximum value of protocol range";
}
}
description
"Filter grouping containing list of Protocol ranges";
}
grouping traffic-group {
    container traffic-group {
        leaf traffic-group-name {
            type string;
            description
            "This leaf defines name of the traffic group referenced";
        }
        description
        "traffic group container";
    }
    description
    "traffic group grouping";
}

augment "/policy:classifiers/policy:classifier" +
"/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";
}
}
description
 "augments diffserv filters to qos classifier";
}
augment "/policy:policies/policy:policy/policy:classifier" + 
  "/policy:filter" {
  when "././policy:type = 'diffserv:ipv4-diffserv-policy-type' or
   ././policy:type = 'diffserv:ipv6-diffserv-policy-type' or
   ././policy:type = 'diffserv:diffserv-policy-type'" {
    description
      "If policy type is v4, v6 or default diffserv,
         this filter can be used.";
  }
}
description
  "Choice of action types";
case dscp {
  uses dscp;
  description
    "Filter containing list of dscp ranges";
}
case source-ipv4-prefix {
  when "/policy:policies/policy:policy/policy:type != " +
    "/diffserv: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 "/policy:policies/policy:policy/policy:type != " +
    "/diffserv: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 "/policy:policies/policy:policy/policy:type != " +
    "/diffserv: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 "/policy:policies/policy:policy/policy:type != " +
    "/diffserv: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";
}
}
description
  "Augments Diffserv Classifier with common filter types";
}
augment "/policy:policies/policy:policy/policy:classifier" + 
  "/policy:action/policy:action-params" {
  when ".//..//policy:type = 'diffserv:ipv4-diffserv-policy-type' or 
    .//..//policy:type = 'diffserv:ipv6-diffserv-policy-type' or 
    .//..//policy:type = 'diffserv: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";

case dscp-marking {
  uses action:dscp-marking;
}
case meter-inline {
  uses action:meter;
}
case meter-reference {
  uses action:meter-reference;
}
case traffic-group-marking {
  uses action:traffic-group-marking;
}
case child-policy {
if-feature action:child-policy-feature;
  uses action:child-policy;
}
case count {
  if-feature action:count-feature;
  uses action:count;
}
case named-count {
  if-feature action:named-counter-feature;
  uses action:named-counter;
}
case queue-inline {
  uses action:queue;
}
case scheduler-inline {
  uses action:scheduler;
}
}

Figure 9: ietf-diffserv module

5.4. ietf-queue-policy

<CODE BEGINS>
file "ietf-queue-policy@2022-07-08.yang"
module ietf-queue-policy {
  yang-version 1.1;
  prefix queue-policy;

  import ietf-traffic-policy {
    prefix policy;
  }
  import ietf-qos-action {
    prefix action;
  }
  import ietf-diffserv {
    prefix diffserv;
  }

  organization
    "IETF Routing Area Working Group";

  contact
    "WG Web:   <https://datatracker.ietf.org/wg/rtgwg/>
    WG List:   <mailto:rtgwg@ietf.org>"
description
"This module contains a collection of YANG definitions for configuring diffserv specification implementations.

Copyright (c) 2021 IETF Trust and the persons identified as authors of the code. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, is permitted pursuant to, and subject to the license terms contained in, the 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 2022-07-08 {
    description
        "Initial version";
    reference
        "RFC XXXX: YANG Models for Quality of Service (QoS)."
}

identity queue-policy-type {
    base policy:policy-type;
    description
        "This defines queue policy-type";
}

augment "/policy:policies/policy:policy/policy:classifier" + "/policy:filter" {
    when ".//.//policy:type = 'queue-policy: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-name {
            uses diffserv:traffic-group;
            description
                "traffic group name";
        }
    }
}

description "Augments Queue policy Classifier with common filter types";)

identity queue-template-name {
  base policy:action-type;
  description "queue template name";
}

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";
}

container queue {
  description "Queue template";
  leaf name {
    type string;
    description "A unique name identifying this queue template";
  }
  uses action:queue;
}
augment "/policy:policies/policy:policy/policy:classifier" + 
  "/policy:action/policy:action-params" {
  when ".//policy:type = 'queue-policy:queue-policy-type'" {
    description "queue policy actions.";
  }
  case queue-template-name {
    uses queue-reference;
  }
  case queue-inline {
    uses action:queue;
  }
}
Figure 10: ietf-queue-policy module

5.5. ietf-scheduler-policy

<CODE BEGINS>
file "ietf-scheduler-policy@2022-07-08.yang"
module ietf-scheduler-policy {
    yang-version 1.1;
    prefix scheduler-policy;

    import ietf-traffic-policy {
        prefix policy;
    }
    import ietf-qos-action {
        prefix action;
    }

    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) 2021 IETF Trust and the persons identified as
    authors of the code. All rights reserved.

    Redistribution and use in source and binary forms, with or
    without modification, is permitted pursuant to, and subject
    to the license terms contained in, the Revised BSD License
    set forth in Section 4.c of the IETF Trust’s Legal Provisions

Choudhary, et al. Expires 8 January 2023 [Page 53]
Internet-Draft             YANG Models for QoS                 July 2022

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 2022-07-08 {
  description
    "Initial version";
  reference
    "RFC XXXX: YANG Models for Quality of Service (QoS).";
}

identity scheduler-policy-type {
  base policy:policy-type;
  description
    "This defines scheduler policy-type";
}

identity filter-match-all {
  base policy:filter-type;
  description
    "Traffic-group filter type";
}

grouping filter-match-all {
  container match-all-cfg {
    leaf match-all-action {
      type empty;
      description
        "match all packets";
    }
    description
      "the match-all action";
  }
  description
    "the match-all filter grouping";
}

augment "/policy:policies/policy:policy/policy:classifier" +
  "/policy:filter" {
  when ".../policy:type =" +
    "/scheduler-policy:scheduler-policy-type" {
    description
      "Only when policy type is scheduler-policy";
  }
  choice filter-params {
    description

"Choice of action types";

case filter-match-all {
    uses filter-match-all;
    description
    "filter match-all";
}

description
"Augments Queue policy Classifier with common filter types";

identity queue-policy-name {
    base policy:action-type;
    description
    "queue policy name";
}

grouping queue-policy-name-cfg {
    container queue-policy-name {
        leaf queue-policy {
            type string;
            mandatory true;
            description
            "This leaf defines name of the queue-policy";
        }
        description
        "container for queue-policy name";
    }
    description
    "queue-policy name grouping";

    augment "/policy:policies/policy:policy/policy:classifier" +
        "/policy:action/policy:action-params" {
        when ".../policy:type =" +
            "scheduler-policy:scheduler-policy-type" {
            description
            "Only when policy type is scheduler-policy";
        }
    case scheduler {
        uses action:scheduler;
    }
    case queue-policy-name {
        uses queue-policy-name-cfg;
    }
    description
    "augments scheduler template reference to scheduler policy";
5.6. ietf-qos-oper

<CODE BEGINS>
file "ietf-qos-oper@2022-07-08.yang"
module ietf-qos-oper {
  yang-version 1.1;
  prefix 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";
  }

  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) 2021 IETF Trust and the persons identified as
     authors of the code. All rights reserved.

     Redistribution and use in source and binary forms, with or
     without modification, is permitted pursuant to, and subject
     to the license terms contained in, the 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)."

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

revision 2022-07-08 {
  description
    "Initial version";
  reference
    "RFC XXXX: YANG Models for Quality of Service (QoS).";
}

identity direction {
  description
    "This is a base identity of traffic direction";
}

identity inbound {
  base direction;
  description
    "Direction of traffic coming into the network entry";
}

identity outbound {
  base direction;
  description
    "Direction of traffic going out of the network entry";
}

grouping named-stats {
  description
    "QoS matching statistics associated with a stats-name";
  leaf pkts {
    type yang:zero-based-counter64;
    description
      "Number of total matched packets associated to a statistics name";
  }
  leaf bytes {
    type yang:zero-based-counter64;
    description
      "Number of total matched bytes associated to 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";
  }
}
grouping meter-stats {
    description
    "Metering counters";
    leaf conform-pkts {
        type yang:zero-based-counter64;
        description
        "Number of packets within conform rate and burst";
    }
    leaf conform-bytes {
        type yang:zero-based-counter64;
        description
        "Total bytes count within conform rate and burst";
    }
    leaf conform-rate {
        type yang:gauge64;
        units "bits-per-second";
        description
        "Traffic Rate measured as conforming";
    }
    leaf exceed-pkts {
        type yang:zero-based-counter64;
        description
        "Number of non-conforming packets which are within peak rate and peak burst";
    }
    leaf exceed-bytes {
        type yang:zero-based-counter64;
        description
        "Total non-conforming bytes count which is within 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 which are beyond peak rate and peak burst";
    }
    leaf violate-bytes {
        type yang:zero-based-counter64;
        description
        "Total bytes count which is beyond peak rate and peak burst";
    }
}
leaf violate-rate {
    type yang:gauge64;
    units "bits-per-second";
    description
        "Traffic Rate measured as violating";
}
leaf meter-drop-pkts {
    type yang:zero-based-counter64;
    description
        "Number of packets dropped by meter";
}
leaf meter-drop-bytes {
    type yang:zero-based-counter64;
    description
        "Bytes of packets dropped by meter";
}
}

grouping classifier-entry-statistics {
    description
        "Statistics for a classifier entry";
    leaf classifier-entry-name {
        type string;
        description
            "Classifier Entry Name";
    }
    leaf classified-pkts {
        type yang:zero-based-counter64;
        description
            "Number of total packets which filtered
to a classifier-entry";
    }
    leaf classified-bytes {
        type yang:zero-based-counter64;
        description
            "Number of total bytes which filtered
to a classifier-entry";
    }
    leaf classified-rate {
        type yang:gauge64;
        units "bits-per-second";
        description
            "Rate of average data flow through a
classifier-entry";
    }
}

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

leaf queue-id {
    type string;
    description
    "Queue Identifier";
}
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 through RED mechanism";
}
leaf red-drop-bytes {
    type yang:zero-based-counter64;
    description
"Total number of bytes dropped through RED mechanism";
}
leaf red-ecn-marked-pkts {
  type yang:zero-based-counter64;
  description
  "Total number of packets ECN marked through RED mechanism";
}
leaf red-ecn-marked-bytes {
  type yang:zero-based-counter64;
  description
  "Total number of bytes ECN marked through RED mechanism";
}
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 through WRED mechanism";
  }
  leaf drop-bytes {
    type yang:zero-based-counter64;
    description
    "Total number of bytes dropped through WRED mechanism";
  }
  leaf ecn-marked-pkts {
    type yang:zero-based-counter64;
    description
    "Total number of packets ECN marked through WRED mechanism";
  }
  leaf ecn-marked-bytes {
    type yang:zero-based-counter64;
    description
    "Total number of bytes ECN marked through WRED mechanism";
  }
}

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

type string;
description
"Meter Identifier";
} 
uses meter-stats;
}

augment "/if:interfaces/if:interface" {

description
"Augments Qos Target Entry to Interface module";
}

container qos-interface-statistics {

cfg false;

description
"Qos Interface statistics";

list stats-per-direction {

description
"Qos Interface statistics for ingress or egress direction";

leaf direction {

type identityref {

base direction;

}

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

leaf policy-name {

type string;

description
"Policy entry 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-statistics {

description
"Classifier Statistics for each Classifier Entry in a Policy applied in a particular direction";

uses classifier-entry-statistics;
}

list named-statistics {

description
"Statistics for a statistics-name";

leaf name {

type string;


description
   "Statistics name";
}

container aggregated {
   description
   "Matched aggregated statistics for a statistics-name";
   uses named-stats;
}

container non-aggregated {
   description
   "Statistics for non-aggregated statistics-name";
   list classifier-statistics {
      description
      "Classifier Statistics for each Classifier Entry in a
       Policy applied in a particular direction";
      uses classifier-entry-statistics;
   }
   list metering-statistics {
      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-statistics {
      description
      "Statistics for each Queue associated with
       the Policy";
      uses queuing-stats;
   }
}

list metering-statistics {
   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-statistics {
   description
   "Statistics for each Queue associated with the Policy";
   uses queueing-stats;
}
6. IANA Considerations

TBD

7. Security Considerations

8. Acknowledgement

The authors wish to thank Ruediger Geib, Fred Baker, Greg Misky, Tom Petch, Acee Lindem, many others for their helpful comments.

MITRE has approved this document for Public Release, Distribution Unlimited, with Public Release Case Number 19-3027.

9. Contributors

The following people have substantially contributed to the editing of this document:

Norm Strahle
Email: nstrahle@juniper.net

10. References

10.1. Normative References


10.2. Informative References


Choudhary, et al. Expires 8 January 2023
Company A, Company B and Company C Diffserv modules augment 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. Queuing and metering counters are realized by augmenting of the QoS target module.

A.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.

```Yang
module example-compa-diffserv {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:example-compa-diffserv";
  prefix example;

  import ietf-traffic-policy {
    prefix policy;
    reference "RFC XXXX: YANG Model for QoS";
  }
  import ietf-qos-action {
    prefix action;
    reference "RFC XXXX: YANG Model for QoS";
  }
  import ietf-diffserv {
    prefix diffserv;
    reference "RFC XXXX: YANG Model for 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 2021-07-12 {
  description
    "Initial revision for diffserv actions on network packets";
  reference
    "RFC 6020: YANG - A Data Modeling Language for the
        Network Configuration Protocol (NETCONF)";
}

identity default-policy-type {
  base policy:policy-type;
  description
    "This defines default policy-type";
}

identity qos-group {
  base policy:filter-type;
  description
    "qos-group filter-type";
}

grouping qos-group-cfg {
  list qos-group-cfg {
    key "qos-group-min qos-group-max";
    description
      "list of dscp ranges";
    leaf qos-group-min {
      type uint8;
      description
        "Minimum value of qos-group range";
    }
    leaf qos-group-max {
      type uint8;
      must ". >= ../qos-group-min" {
        error-message
          "The qos-group-max must be greater than or equal to
              qos-group-min";
      }
  }
}
grouping wred-threshold {
    container wred-min-thresh {
        uses action:threshold;
        description
        "Minimum threshold";
    }
    container wred-max-thresh {
        uses action:threshold;
        description
        "Maximum threshold";
    }
    leaf mark-probability {
        type uint32 {
            range "1..1000";
        }
        description
        "Mark probability";
    }
    description
    "WRED threshold attributes";
}

grouping randomdetect {
    leaf exp-weighting-const {
        type uint32;
        description
        "Exponential weighting constant factor for wred profile";
    }
    uses wred-threshold;
    description
    "Random detect attributes";
}

augment "/policy:classifiers/" +
    "policy:classifier-entry/" +
    "policy:filter-entry/diffserv:filter-param" {
    case qos-group {
        uses qos-group-cfg;
        description
        "maximum value of qos-group range";
    }
    description
    "Filter containing list of qos-group ranges";
}
"Filter containing list of qos-group ranges. Qos-group represent packet metadata information in a device."

description
"augmentation of classifier filters";

augment "/policy:policies/policy:policy-entry/" + 
"policy:classifier-entry/" + 
"policy:classifier-action-entry-cfg/" + 
"policy:action-cfg-params" {
  case random-detect {
    uses randomdetect;
  }
  description
  "Augment the actions to policy entry";
}

augment "/policy:policies" + 
"/policy:policy-entry" + 
"/policy:classifier-entry" + 
"/policy:classifier-action-entry-cfg" + 
"/policy:action-cfg-params" + 
"/diffserv:meter-inline" + 
"/diffserv:meter-type" + 
"/diffserv:one-rate-two-color-meter-type" + 
"/diffserv:one-rate-two-color-meter" + 
"/diffserv:conform-action" + 
"/diffserv:conform-2color-meter-action-params" + 
"/diffserv:conform-2color-meter-action-val" {
  description
  "augment the one-rate-two-color meter conform with actions";
  case meter-action-drop {
    description
    "meter drop";
    uses action:drop;
  }
  case meter-action-mark-dscp {
    description
    "meter action dscp marking";
    uses action:dscp-marking;
  }
}

augment "/policy:policies" + 
"/policy:policy-entry" + 
"/policy:classifier-entry" +
"/policy:classifier-action-entry-cfg" +
"/policy:action-cfg-params" +
"/diffserv:meter-inline" +
"/diffserv:meter-type" +
"/diffserv:one-rate-two-color-meter-type" +
"/diffserv:one-rate-two-color-meter" +
"/diffserv:exceed-action" +
"/diffserv:exceed-2color-meter-action-params" +
"/diffserv:exceed-2color-meter-action-val" {
  description
  "augment the one-rate-two-color meter exceed
  with actions";
  case meter-action-drop {
    description
    "meter drop";
    uses action:drop;
  }
  case meter-action-mark-dscp {
    description
    "meter action dscp marking";
    uses action:dscp-marking;
  }
}

augment "/policy:policies" +
"/policy:policy-entry" +
"/policy:classifier-entry" +
"/policy:classifier-action-entry-cfg" +
"/policy:action-cfg-params" +
"/diffserv:meter-inline" +
"/diffserv:meter-type" +
"/diffserv:one-rate-tri-color-meter-type" +
"/diffserv:one-rate-tri-color-meter" +
"/diffserv:conform-action" +
"/diffserv:conform-3color-meter-action-params" +
"/diffserv:conform-3color-meter-action-val" {
  description
  "augment the one-rate-tri-color meter conform
  with actions";
  case meter-action-drop {
    description
    "meter drop";
    uses action:drop;
  }
  case meter-action-mark-dscp {
    description
    "meter action dscp marking";
uses action:dscp-marking;
}
}

augment "/policy:policies" +
   "/policy:policy-entry" +
   "/policy:classifier-entry" +
   "/policy:classifier-action-entry-cfg" +
   "/policy:action-cfg-params" +
   "/diffserv:meter-inline" +
   "/diffserv:meter-type" +
   "/diffserv:one-rate-tri-color-meter-type" +
   "/diffserv:one-rate-tri-color-meter" +
   "/diffserv:exceed-action" +
   "/diffserv:exceed-3color-meter-action-params" +
   "/diffserv:exceed-3color-meter-action-val" {

description
   "augment the one-rate-tri-color meter exceed
       with actions";

case meter-action-drop {
      description
      "meter drop";
      uses action:drop;
   }

case meter-action-mark-dscp {
      description
      "meter action dscp marking";
      uses action:dscp-marking;
   }
}

augment "/policy:policies" +
   "/policy:policy-entry" +
   "/policy:classifier-entry" +
   "/policy:classifier-action-entry-cfg" +
   "/policy:action-cfg-params" +
   "/diffserv:meter-inline" +
   "/diffserv:meter-type" +
   "/diffserv:one-rate-tri-color-meter-type" +
   "/diffserv:one-rate-tri-color-meter" +
   "/diffserv:violate-action" +
   "/diffserv:violate-3color-meter-action-params" +
   "/diffserv:violate-3color-meter-action-val" {

description
   "augment the one-rate-tri-color meter conform
       with actions";

case meter-action-drop {
      description
      "meter drop";
uses action:drop;
}
case meter-action-mark-dscp {
    description
    "meter action dscp marking";
    uses action:dscp-marking;
}
}

augment "/policy:policies" +
    "/policy:policy-entry" +
    "/policy:classifier-entry" +
    "/policy:classifier-action-entry-cfg" +
    "/policy:action-cfg-params" +
    "/diffserv:meter-inline" +
    "/diffserv:meter-type" +
    "/diffserv:two-rate-tri-color-meter-type" +
    "/diffserv:two-rate-tri-color-meter" +
    "/diffserv:conform-action" +
    "/diffserv:conform-3color-meter-action-params" +
    "/diffserv:conform-3color-meter-action-val" {
    description
    "augment the one-rate-tri-color meter conform with actions";
    case meter-action-drop {
        description
        "meter drop";
        uses action:drop;
    }
    case meter-action-mark-dscp {
        description
        "meter action dscp marking";
        uses action:dscp-marking;
    }
}

augment "/policy:policies" +
    "/policy:policy-entry" +
    "/policy:classifier-entry" +
    "/policy:classifier-action-entry-cfg" +
    "/policy:action-cfg-params" +
    "/diffserv:meter-inline" +
    "/diffserv:meter-type" +
    "/diffserv:two-rate-tri-color-meter-type" +
    "/diffserv:two-rate-tri-color-meter" +
    "/diffserv:exceed-action" +
    "/diffserv:exceed-3color-meter-action-params" +
    "/diffserv:exceed-3color-meter-action-val" {
  description 
  "augment the two-rate-tri-color meter violate with actions";
  case meter-action-drop { 
    description 
    "meter drop";
    uses action:drop;
  }
  case meter-action-mark-dscp { 
    description 
    "meter action dscp marking";
    uses action:dscp-marking;
  }
}
  description 
  "augment the one-rate-two-color meter violate with actions";
  case meter-action-drop { 
    description 
    "meter drop";
    uses action:drop;
  }
  case meter-action-mark-dscp { 
    description 
    "meter action dscp marking";
    uses action:dscp-marking;
  }
}
"/diffserv:one-rate-two-color-meter" {
  description "augment the one-rate-two-color meter with" +
                 "color classifiers";
  container conform-color {
    uses classifier:classifier-entry-generic-attr;
    description "conform color classifier container";
  }
  container exceed-color {
    uses classifier:classifier-entry-generic-attr;
    description "exceed color classifier container";
  }
}

augment "/policy:policies" +
      "/policy:policy-entry" +
      "/policy:classifier-entry" +
      "/policy:classifier-action-entry-cfg" +
      "/policy:action-cfg-params" +
      "/diffserv:meter-inline" +
      "/diffserv:meter-type" +
      "/diffserv:one-rate-tri-color-meter-type" +
      "/diffserv:one-rate-tri-color-meter" {
  description "augment the one-rate-tri-color meter with" +
                 "color classifiers";
  container conform-color {
    uses classifier:classifier-entry-generic-attr;
    description "conform color classifier container";
  }
  container exceed-color {
    uses classifier:classifier-entry-generic-attr;
    description "exceed color classifier container";
  }
  container violate-color {
    uses policy:classifier-entry-generic-attr;
    description "violate color classifier container";
  }
}

augment "/policy:policies" +
      "/policy:policy-entry" +
      "/policy:classifier-entry" +
      "/policy:classifier-action-entry-cfg" +
      "/policy:action-cfg-params" +
  description  
  "augment the two-rate-tri-color meter with" + "color classifiers";  
  container conform-color {  
    uses classifier:classifier-entry-generic-attr;  
    description  
    "conform color classifier container";  
  }  
  container exceed-color {  
    uses policy:classifier-entry-generic-attr;  
    description  
    "exceed color classifier container";  
  }  
  container violate-color {  
    uses policy:classifier-entry-generic-attr;  
    description  
    "violate color classifier container";  
  }  
}  
}

A.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 multiple 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 ietf-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 ietf-traffic-policy {
  prefix policy;
  reference "RFC XXXX: YANG Model for QoS";
}

import ietf-qos-action {
  prefix action;
  reference "RFC XXXX: YANG Model for QoS";
}

import ietf-diffserv {
  prefix diffserv;
  reference "RFC XXXX: YANG Model for QoS";
}

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) 2021 IETF Trust and the persons identified as
  authors of the code.  All rights reserved.
  Redistribution and use in source and binary forms, with or
  without modification, is permitted pursuant to, and subject
  to the license terms contained in, the Simplified BSD License
  set forth in Section 4.c of the IETF Trust's Legal Provisions
  Relating to IETF Documents
  (http://trustee.ietf.org/license-info).
  
  This version of this YANG module is part of RFC XXXX; see
  the RFC itself for full legal notices.";

revision 2021-07-12 {
  description
    "Initial revision of Company B diffserv policy";
  reference "RFC XXXX";
}

/*************************
* Classification types
*/

Choudhary, et al. Expires 8 January 2023 [Page 76]
identity forwarding-class {
  base policy:filter-type;
  description
      "Forwarding class filter type";
}

description
      "Internal loss priority filter type";
}

grouping forwarding-class-cfg {
  list forwarding-class-cfg {
    key "forwarding-class";
    description
      "list of forwarding-classes";
    leaf forwarding-class {
      type string;
      description
        "Forwarding class name";
    }
  }
  description
      "Filter containing list of forwarding classes";
}

grouping loss-priority-cfg {
  list loss-priority-cfg {
    key "loss-priority";
    description
      "list of loss-priorities";
    leaf loss-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";
        }
      }
    }
  }
}

Choudhary, et al. Expires 8 January 2023 [Page 77]
description "Loss-priority";
}
}

description "Filter containing list of loss priorities";

augment "/policy:policies" + 
    "/policy:policy-entry" + 
    "/policy:classifier-entry" + 
    "/policy:filter-entry" + 
    "/diffserv:filter-params" {
    case forwarding-class {
        uses forwarding-class-cfg;
        description "Filter Type Internal-loss-priority";
    }
    case internal-loss-priority {
        uses loss-priority-cfg;
        description "Filter Type Internal-loss-priority";
    }
    description "Augments Diffserv Classifier with vendor" + 
        " specific types";
}

/*************************************************
* Actions                                       
*************************************************/

identity mark-fwd-class {
    base policy:action-type;
    description "mark forwarding class action type";
}

identity mark-loss-priority {
    base policy:action-type;
    description "mark loss-priority action type";
}

grouping mark-fwd-class {
    container mark-fwd-class-cfg {
        leaf forwarding-class {
type string;
description
  "Forwarding class name";
}
description
  "mark-fwd-class container";
}
description
  "mark-fwd-class grouping";
}

grouping mark-loss-priority {
  container mark-loss-priority-cfg {
    leaf loss-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";
        } 
      } 
      description
        "Loss-priority";
    } 
    description
      "mark-loss-priority container";
  } 
  description
    "mark-loss-priority grouping";
}

identity exceed-2color-meter-action-drop {
  base action:exceed-2color-meter-action-type;
  description
    "drop action type in a meter";
}

identity meter-action-mark-fwd-class {
  base action:exceed-2color-meter-action-type;
  description
    "mark forwarding class action type";
identity meter-action-mark-loss-priority {
    base action:exceed-2color-meter-action-type;
    description
        "mark loss-priority action type";
}

identity violate-3color-meter-action-drop {
    base action:violate-3color-meter-action-type;
    description
        "drop action type in a meter";
}

augment "/policy:policies/policy:policy-entry/" +
    "policy:classifier-entry/" +
    "policy:classifier-action-entry-cfg/" +
    "policy:action-cfg-params" {
    case mark-fwd-class {
        uses mark-fwd-class;
        description
            "Mark forwarding class in the packet";
    }
    case mark-loss-priority {
        uses mark-loss-priority;
        description
            "Mark loss priority in the packet";
    }
    case discard {
        uses action:discard;
        description
            "Discard action";
    }
    description
        "Augments common diffserv policy actions";
}

augment "/action:meter-template" +
    "/action:meter-entry" +
    "/action:meter-type" +
    "/action:one-rate-tri-color-meter-type" +
    "/action:one-rate-tri-color-meter" {
    leaf one-rate-color-aware {
        type boolean;
        description
            "This defines if the meter is color-aware";
    }
augment "/action:meter-template" +
  "/action:meter-entry" +
  "/action:meter-type" +
  "/action:two-rate-tri-color-meter-type" +
  "/action:two-rate-tri-color-meter" {
leaf two-rate-color-aware {
  type boolean;
  description
    "This defines if the meter is color-aware";
}
}
/* example of augmenting a meter template with a 
/* vendor specific action */
augment "/action:meter-template" +
  "/action:meter-entry" +
  "/action:meter-type" +
  "/action:one-rate-two-color-meter-type" +
  "/action:one-rate-two-color-meter" +
  "/action:exceed-action" +
  "/action:exceed-2color-meter-action-params" +
  "/action:exceed-2color-meter-action-val" {
  case exceed-2color-meter-action-drop {
    description
      "meter drop";
    uses action:drop;
  }
  case meter-action-mark-fwd-class {
    uses mark-fwd-class;
    description
      "Mark forwarding class in the packet";
  }
  case meter-action-mark-loss-priority {
    uses mark-loss-priority;
    description
      "Mark loss priority in the packet";
  }
}
augment "/action:meter-template" +
  "/action:meter-entry" +
  "/action:meter-type" +
  "/action:two-rate-tri-color-meter-type" +
  "/action:two-rate-tri-color-meter" +
  "/action:violate-action" +
  "/action:violate-3color-meter-action-params" +
  "/action:violate-3color-meter-action-val" {
"/action:violate-3color-meter-action-val" {
  case exceed-3color-meter-action-drop {
    description
    "meter drop";
    uses action:drop;
  }

description
  "Augment the actions to the two-color meter";
}

augment "/action:meter-template" +
  "/action:meter-entry" +
  "/action:meter-type" +
  "/action:one-rate-tri-color-meter-type" +
  "/action:one-rate-tri-color-meter" +
  "/action:violate-action" +
  "/action:violate-3color-meter-action-params" +
  "/action:violate-3color-meter-action-val" {
  case exceed-3color-meter-action-drop {
    description
    "meter drop";
    uses action:drop;
  }

description
  "Augment the actions to basic meter";
}

module example-compb-queue-policy {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:example-compb-queue-policy";
  prefix queue-plcy;

  import ietf-traffic-policy {
    prefix policy;
    reference "RFC XXXX: YANG Model for QoS";
  }

  organization "Company B";
  contact
    "Editor:   XYZ
     <mailto:xyz@compb.com>";

description
  "This module defines a queue policy. The classification
   is based on a forwarding class, and the actions are queues.

Choudhary, et al.        Expires 8 January 2023                [Page 82]"
revision 2021-07-12 {
  description
    "Latest revision of Company B queue policy";
  reference "RFC XXXX";
}

identity forwarding-class {
  base policy:filter-type;
  description
    "Forwarding class filter type";
}

grouping forwarding-class-cfg {
  leaf forwarding-class-cfg {
    type string;
    description
      "forwarding-class name";
  } description
    "Forwarding class filter";
}

augment "/policy:policies" +
  "/policy:policy-entry" +
  "/policy:classifier-entry" +
  "/policy:filter-entry" {
  /* Does NOT support "logical-not" of forwarding class. Use "must"? */
  choice filter-params {
    description
      "Choice of filters";
    case forwarding-class-cfg {
      uses forwarding-class-cfg;
      description
        "Filter Type Internal-loss-priority";
    }
  }
}
description
  "Augments Diffserv Classifier with fwd class filter";
}

identity compb-queue {
  base policy: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 "/policy:policies" +
  "/policy:policy-entry" +
  "/policy:classifier-entry" +
  "/policy:classifier-action-entry-cfg" {
  choice action-cfg-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;
  prefix compb-queue;

  import ietf-qos-action {
    prefix 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 2021-07-12 {
  description
    "Latest revision of diffserv based classifier";
  reference "RFC XXXX";
}

container compb-queue {
  description
    "Queue used in compb architecture";
  leaf name {
    type string;
    description
      "A unique name identifying this queue";
  }
  uses 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-porportion {
          type uint32 {
            range "1..1000";
          }
          description
            "excess-rate-porportion";
        }
      }
    }
  }
}
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-cfg" +
  "/algorithmic-drop-cfg" +
  "/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 action;
    reference "RFC XXXX: YANG Model for QoS";
  }

  import ietf-traffic-policy {
    prefix policy;
    reference "RFC XXXX: YANG Model for 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 2021-07-12 {
    description
      "Initial revision of Company B Scheduler policy";
    reference "RFC XXXX";
  }

  identity queue-policy {
    base policy:action-type;
    description
      "forwarding-class-queue action type";
  }

  grouping queue-policy-name {

Choudhary, et al. Expires 8 January 2023 [Page 88]
container compb-queue-policy-name {
  leaf name {
    type string;
    description
      "Queue policy name";
  }
  description
    "compb-queue-policy container";
  description
    "compb-queue policy grouping";
}

augment "/policy:policies" +
  "/policy:policy-entry" +
  "/policy:classifier-entry" +
  "/policy:classifier-action-entry-cfg" {
  choice action-cfg-params {
    case schedular {
      uses action:schedular;
    }
    case queue-policy {
      uses queue-policy-name;
    }
    description
      "Augment the scheduler policy with a queue policy";
  }
}

A.3. Example of Company C Diffserv Model

Company C vendor augmentation is based on Ericsson’s implementation differentiated QoS. This implementation first sorts traffic based on a classifier, which can sort traffic into one or more traffic forwarding classes. Then, a policer or meter policy references the classifier and its traffic forwarding classes to specify different service levels for each traffic forwarding class.

Because each classifier sorts traffic into one or more traffic forwarding classes, this type of classifier does not align with ietf-qos-classifier.yang, which defines one traffic forwarding class per classifier. Additionally, Company C’s policing and metering policies relies on the classifier’s pre-defined traffic forwarding classes to provide differentiated services, rather than redefining the patterns within a policing or metering policy, as is defined in ietf-diffserv.yang.
Due to these differences, even though Company C uses all the building blocks of classifier and policy, Company C’s augmentation does not use ietf-diffserv.yang to provide differentiated service levels. Instead, Company C’s augmentation uses the basic building blocks, ietf-traffic-policy.yang to provide differentiated services.

```yang
module example-compc-qos-policy {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:example-compc-qos-policy";
  prefix "compcqos";

  import ietf-traffic-policy {
    prefix "pol";
    reference "RFC XXXX: YANG Model for QoS";
  }

  import ietf-qos-action {
    prefix "action";
    reference "RFC XXXX: YANG Model for QoS";
  }

  organization "Company C";
  contact "Company C Editor: XYZ <mailto:xyz@compc.com>";
  description
    "This module contains a collection of YANG definitions for
    configuring different service specification implementations.
    Copyright (c) 2021 IETF Trust and the persons identified as
    authors of the code. All rights reserved.
    Redistribution and use in source and binary forms, with or
    without modification, is permitted pursuant to, and subject
    to the license terms contained in, the Simplified BSD License
    set forth in Section 4.c of the IETF Trust's Legal Provisions
    Relating to IETF Documents
    (http://trustee.ietf.org/license-info).

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

  revision 2021-07-12 {
    description "Initial version";
    reference "RFC XXXX";
  }

  /* identities */

  identity compc-qos-policy {
    base pol:policy-type;
    description "compc-specific policy base type";
  }
}
```

Choudhary, et al. Expires 8 January 2023 [Page 90]
identity mdrq-queuing-policy {
    base compc-qos-policy;
    description "compc-specific MDRR policy type";
}

identity pwfq-queuing-policy {
    base compc-qos-policy;
    description "compc-specific queuing policy type";
}

identity policing-policy {
    base compc-qos-policy;
    description "compc-specific policing policy type";
}

identity metering-policy {
    base compc-qos-policy;
    description "compc-specific metering policy type";
}

identity forwarding-policy {
    base compc-qos-policy;
    description "compc-specific forwarding policy type";
}

identity overhead-profile-policy {
    base compc-qos-policy;
    description "compc-specific overhead profile policy type";
}

identity resource-profile-policy {
    base compc-qos-policy;
    description "compc-specific resource profile policy type";
}

identity protocol-rate-limit-policy {
    base compc-qos-policy;
    description "compc-specific protocol rate limit policy type";
}

identity compc-qos-action {
    base pol:action-type;
    description "compc-specific qos action base type";
}

/* groupings */
grouping redirect-action-grp {
  description "Redirect options grouping";
  container redirect {
    description "Redirect options";
  }
}

/* deviations */

deviation "/pol:policies/pol:policy-entry" {
  deviate add {
    must "pol:type = compc-qos-policy" {
      description "Only policy types drived from compc-qos-policy are supported";
    }
  }
}

deviation "/pol:policies/pol:policy-entry/pol:classifier-entry" {
  deviate add {
    must "./.per-class-action = 'true'" {
      description "Only policies with per-class actions have classifiers";
    }
    must "((./compcqos:sub-type !="compcqos:mdrr-queuing-policy") and " + "(./compcqos:sub-type !="compcqos:pwfq-queuing-policy’)) or " + "((./compcqos:sub-type = "compcqos:mdrr-queuing-policy’) or " + "(./compcqos:sub-type = "compcqos:pwfq-queuing-policy’)) and " + "((classifier-entry-name = '0') or " + "(classifier-entry-name = '1') or " + "(classifier-entry-name = '2') or " + "(classifier-entry-name = '3') or " + "(classifier-entry-name = '4') or " + "(classifier-entry-name = '5') or " + "(classifier-entry-name = '6') or " + "(classifier-entry-name = '7') or " + "(classifier-entry-name = '8'))" {
      description "MDRR queuing policy’s or PWFP queuing policy’s " + "classifier-entry-name is limited to the listed values";
    }
  }
}
deviation "/pol:policies/pol:policy-entry/pol:classifier-entry" + 
"/pol:classifier-action-entry-cfg" {
  deviate add {
    must "action-type = 'compcqos:compc-qos-action'" {
      description
      "Only compc-qos-action is allowed";
    }
    max-elements 1;
  }
}

/* augments */

augment "/pol:policies/pol:policy-entry" {
  when "pol:policy-type = 'compc-qos-policy'" {
    description
    "Additional nodes only for diffserv-policy";
  }
  description "Additional diffserv-policy nodes";
  leaf sub-type {
    type identityref {
      base compc-qos-policy;
    }
    mandatory true;
    description "Policy sub-type. The value of this leaf must " + 
    "not change once configured";
  }
  leaf per-class-action {
    type boolean;
    must "((. = 'true') and " + 
    "((../compcqos:sub-type = " + 
      "'compcqos:policing-policy') or " + 
      "(../compcqos:sub-type = " + 
      "'compcqos:metering-policy') or " + 
      "(../compcqos:sub-type = " + 
      "'compcqos:mdrr-queuing-policy') or " + 
      "(../compcqos:sub-type = " + 
      "'compcqos:pwfq-queuing-policy') or " + 
      "(../compcqos:sub-type = " + 
      "'compcqos:forwarding-policy')))) or " + 
    "((. = 'false') and " + 
    "((../compcqos:sub-type = " + 
      "'compcqos:overhead-profile-policy') or " + 
      "(../compcqos:sub-type = " + 
      "'compcqos:resource-profile-policy') or " + 
      "(../compcqos:sub-type = " + 
      "'compcqos:protocol-rate-limit-policy'))))" {
      description
      "";
    }
  }
}
"Only certain policies have per-class action;"
}
mandatory true;
description "Per-class action;"
}
container traffic-classifier {
  when "../compcqos:sub-type = 'compcqos:policing-policy' or " + 
    "../compcqos:sub-type = 'compcqos:metering-policy' or " + 
    "../compcqos:sub-type = 'compcqos:forwarding-policy'"
  {
    description
    "A classifier for policing-policy or metering-policy;"
  }
presence true;
leaf name {
type string;
  mandatory true;
description
  "Traffic classifier name;"
}
leaf type {
type enumeration {
  enum 'internal-dscp-only-classifier' {
    value 0;
description
    "Classify traffic based on (internal) dscp only;"
  }
  enum 'ipv4-header-based-classifier' {
    value 1;
description
    "Classify traffic based on IPv4 packet header fields;"
  }
  enum 'ipv6-header-based-classifier' {
    value 2;
description
    "Classify traffic based on IPv6 packet header fields;"
  }
}
  mandatory true;
description
  "Traffic classifier type;"
}
description "Traffic classifier;"
}
container traffic-queue {
  when "(.compcqos:sub-type = " + 
    "compcqos:mdrr-queuing-policy") or " + 
    "(.compcqos:sub-type = " + 
    "compcqos:pwfq-queuing-policy")"
  {

description
  "Queuing policy properties";
}
leaf queue-map {
  type string;
  description
    "Traffic queue map for queuing policy";
}  

description "Traffic queue";
}
container overhead-profile {
  when ".//compcqos:sub-type = " +
        "/compcqos:overhead-profile-policy" " {
    description
      "Overhead profile policy properties";
  }  
  description "Overhead profile";
}
container resource-profile {
  when ".//compcqos:sub-type = " +
        "/compcqos:resource-profile-policy" " {
    description
      "Resource profile policy properties";
  }  
  description "Resource profile";
}
container protocol-rate-limit {
  when ".//compcqos:sub-type = " +
        "/compcqos:protocol-rate-limit-policy" " {
    description
      "Protocol rate limit policy properties";
  }  
  description "Protocol rate limit";
}
}

augment "/pol:policies/pol:policy-entry/pol:classifier-entry" +
   "/pol:classifier-action-entry-cfg/pol:action-cfg-params" {
  when ".//pol:policy-type = 'compc-qos-policy' " {
    description
      "Configurations for a classifier-policy-type policy";
  }  
  case metering-or-policing-policy {
    when ".//compcqos:sub-type = " +
        "/compcqos:policing-policy' or " +
        ".//compcqos:sub-type = 'compcqos:metering-policy'" {
      container dscp-marking {  

uses action:dscp-marking;
    description "DSCP marking";
}

container precedence-marking {
    uses action:dscp-marking;
    description "Precedence marking";
}

container priority-marking {
    uses action:prioriry;
    description "Priority marking";
}

container rate-limiting {
    uses action:one-rate-two-color-meter;
    description "Rate limiting";
}

}  

}  

}  

}  

{  
  when "../../compcqos:sub-type = " +
     byss:compcqos:mdrr-queuing-policy’" {  
    description
      "MDRR queue handling properties for the traffic " +
      "classified into current queue";
  }

  leaf mdrr-queue-weight {
    type uint8 {
      range "20..100";
    }
    units percentage;
    description "MDRR queue weight";
  }

}  

}  

}  

{  
  when "../../compcqos:sub-type = " +
     byss:compcqos:pwfq-queuing-policy’" {  
    description
      "PWFQ queue handling properties for traffic " +
      "classified into current queue";
  }

  leaf pwfq-queue-weight {
    type uint8 {
      range "20..100";
    }
    units percentage;
    description "Priority-based weighted fair queue weight";
  }

  leaf pwfq-queue-priority {
    type uint8;
    description "Priority-based weighted fair queue priority";
  }
A.4. Configuration example for QoS Classifier

<!--
This example shows a QoS classifier configuration.
-->

<?xml version="1.0" encoding="UTF-8"?>
<classifiers
  xmlns="urn:ietf:params:xml:ns:yang:ietf-traffic-policy"
  xmlns:bt="urn:ietf:params:xml:ns:yang:ietf-bgp-types">
  <classifier>
    <name>my-classifier</name>
    <filter-operation>match-any-filter</filter-operation>
    <filter>
      <type>dscp</type>
      <logical-not>true</logical-not>
    </filter>
  </classifier>
</classifiers>

Figure 13: Configuration example for QoS Classifier

A.5. 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"><policy><name>my-policy</name><type>diffserv</type><classifier><name>my-classifier</name><inline>true</inline><filter-operation>match-any-filter</filter-operation><filter><type>dscp</type><logical-not>true</logical-not></filter><action><type>dscp-marking</type><!-- Add the action-params here once it has been defined. --></action></classifier></policy></policies>

Figure 14: Configuration example for QoS Policy

Authors’ Addresses

Aseem Choudhary
Aviatrix Systems
2901 Tasman Drive #109
Santa Clara, CA 95054
United States of America
Email: asechoud@cisco.com

Mahesh Jethanandani
Kloud Services
Email: mjethanandani@gmail.com
Problems and Requirements of Satellite Constellation for Internet
draft-lhan-problems-requirements-satellite-net-03

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

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 7 January 2023.
Table of Contents

1. Introduction ...................................................... 3
2. Terminology ..................................................... 3
3. Overview .......................................................... 5
4. Basics of Satellite Constellation ................................. 6
   4.1. Satellite Orbit .............................................. 6
   4.2. Coverage of LEO and VLEO Satellites and Minimum Number
        Required .................................................. 7
   4.3. Real Deployment of LEO and VLEO for Satellite Network ........ 9
5. Communications for Satellite Constellation ......................... 10
   5.1. Dynamic Ground-station-Satellite Communication ................ 11
   5.2. Dynamic Inter-satellite Communication ........................ 12
      5.2.1. Inter-satellite Communication Overview ................ 12
      5.2.2. Satellites on Adjacent Orbit Planes with Same
             Altitude ............................................... 15
      5.2.3. Satellites on Adjacent Orbit Planes with Different
             Altitude ............................................... 17
6. Use Satellite Network for Internet Integration ..................... 19
   6.1. Use Satellite Network for Broadband Access ................... 20
   6.2. Use Satellite Network with 3GPP Wireless Access
        Network .................................................... 22
   6.3. Recent Development and Study in 3GPP for Satellite
        Network .................................................... 22
7. Problems and Requirements for Satellite Constellation for
   Internet .................................................................. 25
   7.1. Common Problems and Requirements ............................. 25
   7.2. Satellite Relay ................................................ 27
      7.2.1. One Satellite Relay .................................... 28
      7.2.2. Multiple Satellite Relay ................................. 29
   7.3. Satellite Networking .......................................... 31
      7.3.1. L2 or L3 network ......................................... 31
      7.3.2. Inter-satellite-Link Lifetime ............................ 31
      7.3.3. Problems for Traditional Routing Technologies ........... 32
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
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
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. 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. 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 not relatively stationary to the ground. This will lead to:

1. More satellites than GSO are needed to provide the global coverage. Section 4.2 will analyze the coverage area, and the minimum number of satellites required to cover the earth surface.
2. The point-to-point communication between satellite and ground station will not be static. Mobility issue has to be considered. Detailed analysis will be done in Section 5.1.

3. The inter-satellite communication is needed, and all satellites need to form a network. Details are described in Section 5.2.

In addition to above context, Section 7 will address the problem and requirements when satellite constellation is joining Internet.

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 public. Authors of the draft have no relationship or relevant inside knowledge of SpaceX/Starlink.

4. 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.

4.1. Satellite Orbit

The orbit of a satellite can be either circular or elliptic, 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.
4.2. 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 1 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 2 shows how to calculate the radius (\(R_c\)) of coverage area from the satellite altitude (\(A_s\)) and the elevation angle (\(b\)).
Figure 1: Satellite coverage on ground

|<--- 2*Rc --->|

+ Satellite
/|
/ | b
/\___Earth surface
/ * | |
/ *__________ |
+ | | |
* 2*a * * |
* ___ * * |
*-- * * |
* * * |
* Earth center

Figure 2: 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.  \( Rc = Re \times (a \pi) / 180 \)

a  The cap angle for the coverage area (the RC arc).  \( a = \arccos((Re/(Re+As)) \times \cos(b)) - b \).

b  The least elevation angle that a ground station or a terminal can communicate with a satellite, \( b = 35 \text{ degree} \).

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, \( Ns = 180/(a \times \cos(30)) \)

No  The minimum number of orbit (with same inclination), it is equal to the number of the satellite orbit’s vertical projection, so, \( No = 360/(a \times (1 + \sin(30))) \)

For a example of two type of satellite LEO and VEO, the coverages are calculated as in Table 1:

<table>
<thead>
<tr>
<th>Parameters</th>
<th>VLEO1</th>
<th>VLEO2</th>
<th>LEO1</th>
<th>LEO2</th>
</tr>
</thead>
<tbody>
<tr>
<td>As (km)</td>
<td>335.9</td>
<td>450</td>
<td>1100</td>
<td>1150</td>
</tr>
<tr>
<td>a (degree)</td>
<td>3.907</td>
<td>5.078</td>
<td>10.681</td>
<td>11.051</td>
</tr>
<tr>
<td>Rc (km)</td>
<td>435</td>
<td>565</td>
<td>1189</td>
<td>1230</td>
</tr>
<tr>
<td>Ns</td>
<td>54</td>
<td>41</td>
<td>20</td>
<td>19</td>
</tr>
<tr>
<td>No</td>
<td>62</td>
<td>48</td>
<td>23</td>
<td>22</td>
</tr>
</tbody>
</table>

Table 1: Satellite coverage estimation for LEO and VEO examples

4.3. Real Deployment of LEO and VEO 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 redundancy 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].

5. 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.
5.1. 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 1.

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/180
SD  The space distance(in km) of two neighbor satellite on the same orbit 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
<table>
<thead>
<tr>
<th>Parameters</th>
<th>VLEO1</th>
<th>VLEO2</th>
<th>LEO1</th>
<th>LEO2</th>
<th>LEO3</th>
</tr>
</thead>
<tbody>
<tr>
<td>As (km)</td>
<td>335.9</td>
<td>450</td>
<td>1100</td>
<td>1150</td>
<td>1325</td>
</tr>
<tr>
<td>a (degree)</td>
<td>3.907</td>
<td>5.078</td>
<td>10.681</td>
<td>11.051</td>
<td>12.293</td>
</tr>
<tr>
<td>AL (km)</td>
<td>793</td>
<td>1048</td>
<td>2415</td>
<td>2515</td>
<td>2863</td>
</tr>
<tr>
<td>SD (km)</td>
<td>792.5</td>
<td>1047.2</td>
<td>2404</td>
<td>2503.2</td>
<td>2846.1</td>
</tr>
<tr>
<td>V (km/s)</td>
<td>7.7</td>
<td>7.636</td>
<td>7.296</td>
<td>7.272</td>
<td>7.189</td>
</tr>
<tr>
<td>T (s)</td>
<td>103</td>
<td>137</td>
<td>331</td>
<td>346</td>
<td>398</td>
</tr>
</tbody>
</table>

Table 2: The time for the ground-station-satellite communication

5.2. Dynamic Inter-satellite Communication

5.2.1. 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 ceiling 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 3 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
 | s1 |
 s4  s6
 | s2  -------- Equator
s5  s7
 | s3 |
s  s
\ s /
 s  s
 | s |
\  |
 + South pole
```

Figure 3: 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 3, s2 can communicate 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 4 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 Section 5.2.2.

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 5 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 Section 5.2.3
* The total number of orbit planes are \( N \)
* The number \((i-1, i, i+1, \ldots)\) represents the Orbit index
* The bottom numbers \((i-1, i, i+1)\) are for orbit planes on which satellites \((S_1, S_2, S_3)\) 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 \((S_4, S_5, S_6)\) are moving from up to bottom.

Figure 4: Two satellites with same altitude and inclination \((i)\) move in the same or opposite direction

* Two satellites \(S_1\) and \(S_2\) are at position \(P_1\) and \(P_2\) at time \(T_1\)
* \(S_1\)’s right facing ISL connected to \(S_2\)’s left facing ISL
* \(S_1\) and \(S_2\) move to the position \(P_4\) and \(P_3\) at time \(T_2\)
* \(S_1\)’s left facing ISL connected to \(S_2\)’s right facing ISL

Figure 5: Two satellites meeting in the polar area will change its facing of ISL

5.2.2. 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 6 illustrates a general case that two satellites move and intersect with an angle \(A\).
More specifically, for orbit planes with the inclination angle $i$, Figure 7 illustrates two satellites move in the opposite direction and intersect with an angle $2i$.

\[ \text{Figure 7: Two satellites with same altitude and inclination (i) intersect with angle } A=2i \]

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.

- $D_1$ The laser communication limit, $D_1=2000\text{km}$
- $A$ The angle between two orbit’s vertical projection on Earth. $A=2i$
- $V_1$ The speed vector of satellite on orbit1
- $V_2$ The speed vector of satellite on orbit2
- $|V|$ the magnitude of the difference of two speed vector $V_1$ and $V_2$, $|V|=|V_1-V_2|=$sqrt{$(V_1-V_2*\cos(A))^2+(V_2*\sin(A))^2$}. For satellites with the same altitude and inclination angle $i$, $V_1=V_2$, so, $|V|=V_1*\text{sqrt}(2-2*\cos(2i))=2V_1*\sin(i)$
The lifetime two satellites can communicate, or the time of two satellites’ distance is within the range of communication, \( T = \frac{2 \cdot D_l}{|V|} \).

| 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)

5.2.3. 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 8 illustrates two satellites (with the altitude difference \( D_a \)) move and intersect with an angle \( A \).

\[
\begin{align*}
^V V2 \\
/ \\
/ \\
-------/ \\
Da / +-- \\
/ / \ A \\
---------/------------ V1 \\
/ \\
/ \\
/ \\
\end{align*}
\]

Figure 8: Satellite (speed vector \( V1 \) and \( V2 \), Altitude difference \( D_a \)) intersects with Angle \( A \)

Follows are the math to calculate the lifetime of communication

\( D_l \) The laser communication limit, \( D_l=2000\text{km} \)

\( D_a \) Altitude difference (in km) for two orbit planes
A  The angle between two orbit’s vertical projection on Earth

V1  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).

<table>
<thead>
<tr>
<th>Parameters</th>
<th>VLEO1</th>
<th>VLEO2</th>
</tr>
</thead>
<tbody>
<tr>
<td>As(km)</td>
<td>335.9</td>
<td>450</td>
</tr>
<tr>
<td>V (km/s)</td>
<td>7.7</td>
<td>7.636</td>
</tr>
</tbody>
</table>

Table 4: Two VLEO with different altitude and speed

<table>
<thead>
<tr>
<th>A (degree)</th>
<th>0</th>
<th>10</th>
<th>45</th>
<th>90</th>
<th>135</th>
<th>180</th>
</tr>
</thead>
<tbody>
<tr>
<td>V (km/s)</td>
<td>0.065</td>
<td>1.338</td>
<td>5.869</td>
<td>10.844</td>
<td>14.169</td>
<td>15.336</td>
</tr>
<tr>
<td>T(s)</td>
<td>61810</td>
<td>2984</td>
<td>680</td>
<td>368</td>
<td>282</td>
<td>260</td>
</tr>
</tbody>
</table>

Table 5: Two VLEO intersects with different angle and the life of communication
### Table 6: Two LEO with different altitude and speed

<table>
<thead>
<tr>
<th>Parameters</th>
<th>LEO1</th>
<th>LEO2</th>
</tr>
</thead>
<tbody>
<tr>
<td>As (km)</td>
<td>1150</td>
<td>1325</td>
</tr>
<tr>
<td>V (km/s)</td>
<td>7.272</td>
<td>7.189</td>
</tr>
</tbody>
</table>

### Table 7: Two LEO intersects with different angle and the life of communication

<table>
<thead>
<tr>
<th>A (degree)</th>
<th>0</th>
<th>10</th>
<th>45</th>
<th>90</th>
<th>135</th>
<th>180</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>V</td>
<td>(km/s)</td>
<td>0.083</td>
<td>1.263</td>
<td>5.535</td>
<td>10.226</td>
</tr>
<tr>
<td>T(s)</td>
<td>47961</td>
<td>3155</td>
<td>720</td>
<td>390</td>
<td>298</td>
<td>276</td>
</tr>
</tbody>
</table>

6. Use Satellite Network for Internet Integration

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.

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 6.1), another is for the integration with 3GPP wireless network including 4G and 5G (see Section 6.2 and Section 6.3).

6.1. 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 9 for one satellite relay, Figure 10 for multiple satellite relay).

2. The end user terminal access Internet through inter-satellite-networking (Figure 11).

3. The local network access Internet through satellite relay (Figure 12 for one satellite relay, Figure 13 for multiple satellite relay).

4. The local network access Internet through inter-satellite-networking (Figure 14).

```
S1----\         /----------/  
 \ /          /           
 T---GW--GS1--S2--GS2-------PE   Internet   +
 \ /          /           
 \---S3/      \-----------/  
```

Figure 9: End user terminal access Internet through one satellite relay
In above Figure 9 to Figure 14, 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

6.2. 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 15 for mobile access network to satellite network, and Figure 16 for satellite network to mobile access network.

```
+--------------+    +-------------+    +---------+    +--------+
|    T or      |    |Mobile Access|    |Satellite|    |Internet|
| Local network|----+  Network    |----+ Network +----+        |
+--------------+    +-------------+    +---------+    +--------+
```

Figure 15: 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 16: End user terminal or local network access Internet through Satellite Network and Mobile Access Network

6.3. Recent Development and Study in 3GPP for Satellite Network

3GPP SA Working Groups (WG) feature a couple of satellite-related projects (or SIDs). The SA2 WG is currently studying the adoption of satellite communication to provide 5G backhaul service [TR-23.700-27].
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 17 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#.. |
----------+       +--------+
\                                   :                        v
\                                   :(N19)      / \                  \ ^
\                                   :(ISL)    GS ---+ 5G Core +
\                                   :                        ^
\                                   |
\                                   |
\                                   +----------+       +--------+
\                                   | UPF#2   | (ISL) | Sat#.. |
\                                   +----------+       +--------+
| UE#2--gNB#2--| (on-board) | - - - | |
\                                   Sat#2

Figure 17: 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.

7. Problems and Requirements for Satellite Constellation for Internet

As described in Section 6, 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 6, following contexts only discuss the most plausible solution from networking perspectives.

Section 7.1 will list the common problems and requirements for both satellite relay and satellite networking.

Section 7.2 and Section 7.3 will describe key problems, requirement and potential solution from the networking perspective for these two cases respectively.

7.1. Common Problems and Requirements

For both satellite relay and satellite networking, satellite-ground-station 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 9 illustrates this case. Normally microwave is used for both links.

Additionally, from the coverage analysis in Section 4.2 and real deployment in Section 4.3, 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 Section 5.1, 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 Section 5.2.2, 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 data 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.

7.2. 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.
7.2.1. 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.

7.2.2. 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 7.2.1.

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 Section 5.1, 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 7.2.1

In addition to the above requirements, the problem P1-1 in Section 7.2.1 should also apply.

S2. Automatic decision by routing protocol.

This solution is only feasible after the IP continuity problem (P1-1 in Section 7.2.1) 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 7.2.1

In addition to the above requirements, the problem P1-1 in Section 7.2.1 should also apply.
7.3. Satellite Networking

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.

7.3.1. L2 or L3 network

The 1st question to answer is should the satellite network be configured as L2 or L3 network? As analyzed in Section 4.2 and Section 4.3, 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.

7.3.2. 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 Section 5.1. 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 Section 5.2.2 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 Section 5.2, 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 7.3.3.

7.3.3. Problems for Traditional Routing Technologies

When the satellite network is integrated with Internet by traditional routing technologies, following provisioning and configuration (see Figure 18) 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.
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 Section 5.1. 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-convergence, 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-convergence, 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 Section 5.1 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.

8. IANA Considerations

This memo includes no request to IANA.
9. Contributors

10. Acknowledgements

11. References

11.1. Normative References


11.2. Informative References


[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",

[BGP-Converge-Time2]
"Bringing SDN to the Internet, one exchange point at the time",

[StarLink-Ground-Station-Fcc]
"APPLICATION FOR BLANKET LICENSED EARTH STATIONS",

[SpaceX-Ground-Station-Fcc]
"List of SpaceX applications for ground stations",

[Tech-Comparison-LEOs]
"A Technical Comparison of Three Low Earth Orbit Satellite Constellation Systems to Provide Global Broadband",

[StarLink-GW-GS-map]
"Starlink gateway ground station map",
<https://www.google.com/maps/d/u/0/viewer?mid=1H1x8jJ2s8vfy60TvKgpYs_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",
May 2022.

[TS-23.501] 3GPP, "System architecture for the 5G System (5GS)",
December 2021.
Appendix A. Change Log

* Initial version, 07/03/2021
* 01 version, 10/20/2021
* 02 version, 2/13/2022
* 03 version, 7/5/2022

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 McCarthy Blvd.
Milpitas, CA 95035,
United States of America
Email: tianjijiang@chinamobile.com

Ning Wang
University of Surrey
Guildford
Surrey, GU2 7XH
United Kingdom
Email: n.wang@surrey.ac.uk
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.
Table of Contents

1. Introduction ............................................ 3
2. Terminology ............................................. 4
3. Review of LEO satellite constellation for future Internet .. 5
4. Basics of Instructive Routing ............................. 6
5. IPv6 Routing Header for Instructive Routing ............... 9
6. Instruction List for Instructive Routing .................. 10
7. Instructive Routing Behaviors ............................ 11
   7.1. Fwd.Inc.Sat_ID ................................... 12
   7.2. Fwd.Dec.Sat_ID ................................... 13
   7.3. Fwd.Inc.Opb_ID ................................... 14
   7.4. Fwd.Dec.Opb_ID ................................... 15
   7.5. Fwd.Inc.Shl_ID ................................... 16
   7.6. Fwd.Dec.Shl_ID ................................... 17
   7.7. End.Intf_ID ....................................... 18
   7.8. End.Punt ........................................... 18
   7.9. End.Lookup ........................................ 19
   7.10. End.Lookup.IPv4 ................................... 19
   7.11. End.Lookup.IPv6 ................................... 20
   7.12. Fwd.Sat_Addr ..................................... 20
   7.13. Fwd.Sat_MacAddr .................................. 21
   7.15. Forwarding_API_SAT(Packet,Input_Satellite,Sat_Addr) .. 22
   7.16. Forwarding_API_MAC(Packet,Input_Satellite,Sat_MacAddr) . 22
7.17. Forwarding_GS_API(Packet,Input_Interface) ............. 23
8. IANA Considerations ...................................... 23
9. Security Considerations .................................. 23
10. Contributors ............................................ 24
11. Acknowledgements ...................................... 24
12. References ............................................. 24
12.1. Normative References ................................ 24
12.2. Informative References ............................... 24
Appendix A. Change Log ..................................... 25
Authors’ Addresses ......................................... 25
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 lifetime. [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 satellite network, it is based on the satellite semantic address. The routing information is embedded into the IPv6 packet as routing extension header defined in [RFC8200]. Unlike the segment routing [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, 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 calculation for a LEO constellation.
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
Internet-Draft Satellite Instructive Routing March 2022

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. 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.

4. Basics of Instructive Routing

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
The un-steady link can connect to any satellite but only last for a short time. The usage 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

Figure 2 illustrates a 2D example. It shows how a packet is forwarded in a grid satellite network. The forwarding path consists of a series of segments, and each segment consists of two satellites at its two ends. One segment could be on either the same orbit plane or crossing adjacent orbit plane. 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

Obviously, at each satellite, the forwarding logic needs to check if the satellite reaches the end of a segment on the route path. In the regular segment routing, the SID is used to do such indication. But for satellite network, since satellite’s semantic address is embedded into the IPv6 address, it is not needed to include the long SID into the packet header. Instead, it will be much saving if we only embed one of three indexes information of the satellite semantic address in the instruction argument, and then 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)
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 RoutingHdr 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 a 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 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
\--------------/--------------/ \--------------/--------------/
instruction[0]                   instruction[1]...  
```

Figure 4: The Instruction List for Instructive Routing

Funct. 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.

<table>
<thead>
<tr>
<th>Func Name/Hex Value</th>
<th>Arguments/Size(Octet)</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fwd.Inc.Sat_ID/0X01</td>
<td>Sat_ID/1</td>
<td>Section 7.1</td>
</tr>
<tr>
<td>Fwd.Dec.Sat_ID/0X02</td>
<td>Sat_ID/1</td>
<td>Section 7.2</td>
</tr>
<tr>
<td>Fwd.Inc.Obp_ID/0X03</td>
<td>Obp_ID/1</td>
<td>Section 7.3</td>
</tr>
<tr>
<td>Fwd.Dec.Obp_ID/0X04</td>
<td>Obp_ID/1</td>
<td>Section 7.4</td>
</tr>
<tr>
<td>Fwd.Inc.Shl_ID/0X05</td>
<td>Shl_ID/1</td>
<td>Section 7.5</td>
</tr>
<tr>
<td>Fwd.Dec.Shl_ID/0X06</td>
<td>Shl_ID/1</td>
<td>Section 7.6</td>
</tr>
<tr>
<td>End.Intf_ID/0X07</td>
<td>Intf_ID/1</td>
<td>Section 7.7</td>
</tr>
<tr>
<td>End.Punt/0X08</td>
<td>0X0/1</td>
<td>Section 7.8</td>
</tr>
<tr>
<td>End.Lookup/0X09</td>
<td>0X0/1</td>
<td>Section 7.9</td>
</tr>
<tr>
<td>End.Lookup.IPv4/0X0A</td>
<td>IPv4_Addr/4</td>
<td>Section 7.10</td>
</tr>
<tr>
<td>End.Lookup.IPv6/0X0B</td>
<td>IPv6_Addr/16</td>
<td>Section 7.11</td>
</tr>
<tr>
<td>Fwd.Sat_Addr/0X0C</td>
<td>Sat_Addr/4</td>
<td>Section 7.12</td>
</tr>
<tr>
<td>Fwd.Sat_MacAddr/0X0D</td>
<td>Sat_MacAddr/6</td>
<td>Section 7.13</td>
</tr>
</tbody>
</table>

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.
When an IRH is processed {
    If ((RI > 1) and (Argument != Shl_index)) {
        Input_Satellite = Current Satellite;
        Input_Direction = Orbit Shell Index Decremental direction;
        Forwarding_API(Packet, Input_Satellite, Input_Direction);
    } else {
        IOF += 2;
        RI --;
        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;
        Proceed to execute the next Instruction;
    }
}

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.

When an IRH is processed {
    Change the Next header in the packet header to be
    the Next Header field in the Instructive Routing header;
    Remove the Instructive Routing Header;
    Forwarding_GS_API(Packet, Argument);
}

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.}

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. 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
values for the subtype registries are given in Table 1.

9. 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.
10. Contributors

11. Acknowledgements

12. References

12.1. Normative References


12.2. Informative References


Appendix A. Change Log

* Initial version, 02/28/2022

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
Satellite Semantic Addressing for Satellite Constellation
draft-lhan-satellite-semantic-addressing-01

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 7 September 2022.

Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (https://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights
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]
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 steady 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.

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. Same shell group of satellite will have the same altitude and inclination.

3. The total N 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/N). 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 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.

5. All satellites in the same shell group are moving in the same circular direction within the same orbit plane. 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
/     /     /
/     /     /
/     /     /
/orbit /     orbit
/orbit /     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].
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.

Figure 3 and Figure 4 illustrate three types of indexes for satellite.
Figure 3: Shell Group and Orbit Plane Indexes for Satellites

Shell Group and Orbit Plane Indexes for Satellites

Figure 4

Three type 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. Spacelink 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.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>VLEO1</th>
<th>VLEO2</th>
<th>LEO1</th>
<th>LEO2</th>
<th>LEO3</th>
<th>LEO4</th>
<th>LEO5</th>
</tr>
</thead>
<tbody>
<tr>
<td>As(km)</td>
<td>160</td>
<td>300</td>
<td>600</td>
<td>900</td>
<td>1200</td>
<td>1500</td>
<td>2000</td>
</tr>
<tr>
<td>Rc(km)</td>
<td>318</td>
<td>562</td>
<td>1009</td>
<td>1382</td>
<td>1702</td>
<td>1981</td>
<td>2379</td>
</tr>
<tr>
<td>Ns</td>
<td>73</td>
<td>42</td>
<td>23</td>
<td>17</td>
<td>14</td>
<td>12</td>
<td>10</td>
</tr>
<tr>
<td>No</td>
<td>85</td>
<td>48</td>
<td>27</td>
<td>20</td>
<td>16</td>
<td>14</td>
<td>12</td>
</tr>
</tbody>
</table>

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
<table>
<thead>
<tr>
<th>Parameters</th>
<th>VLEO1</th>
<th>VLEO2</th>
<th>LEO1</th>
<th>LEO2</th>
<th>LEO3</th>
<th>LEO4</th>
<th>LEO5</th>
</tr>
</thead>
<tbody>
<tr>
<td>As(km)</td>
<td>160</td>
<td>300</td>
<td>600</td>
<td>900</td>
<td>1200</td>
<td>1500</td>
<td>2000</td>
</tr>
<tr>
<td>Rc(km)</td>
<td>218</td>
<td>392</td>
<td>726</td>
<td>1015</td>
<td>1271</td>
<td>1498</td>
<td>1828</td>
</tr>
<tr>
<td>Ns</td>
<td>107</td>
<td>59</td>
<td>32</td>
<td>23</td>
<td>19</td>
<td>16</td>
<td>13</td>
</tr>
<tr>
<td>No</td>
<td>123</td>
<td>69</td>
<td>37</td>
<td>27</td>
<td>22</td>
<td>18</td>
<td>15</td>
</tr>
</tbody>
</table>

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 limit of LEO orbits and the cost of satellite
constellations, the total number of satellite constellation is
very limited. So, the size of code is limited.

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. When satellite network is using L3 solution, the satellite
semantic address is encoded as the interface identifier (i.e.,
the rightmost 64 bits) of the IPv6 address for IPv6. Figure 5
shows the format of IPv6 Satellite Address.

2. 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 6. 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.

3. 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 7 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
˜  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 5: The IPv6 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:
Internet-Draft        Satellite Semantic Addressing           March 2022

* 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| 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.  IANA Considerations

This memo may include request to IANA for owner code, see
Section 5.4.

7.  Contributors

8.  Acknowledgements

9.  References

9.1.  Normative References

Border Gateway Protocol 4 (BGP-4)", RFC 4271,
DOI 10.17487/RFC4271, January 2006,

Han, et al.             Expires 7 September 2022               [Page 18]
9.2. Informative References

[I-D.lhan-problems-requirements-satellite-net]

[I-D.jia-flex-ip-address-structure]

[I-D.li-native-short-addresses]
[Routing-for-LEO]

[Length-Variable-IP]

[IEEE-MAC-Address]

[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]

[OSI-Model]


[China-constellation]

[SpaceX-Non-GEO]
Appendix A. Change Log

* Initial version, 10/19/2021
* 01 version, 02/28/2022

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
Abstract

This note specifies using packets’ source addresses in route lookups as additional qualifier to be used in hop-by-hop routing decisions. This applies to IPv6 [RFC2460] in general with specific considerations for routing protocol left for separate documents. There is nothing precluding similar operation in IPv4, but this is not in scope of this document.

Note that destination/source routing, source/destination routing, SADR, source-specific routing, source-sensitive routing, S/D routing and D/S routing are all used synonymously.

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 6 January 2023.

Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.
This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (https://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction .................................................. 3
   1.1. Requirements Language ................................. 3
2. Use cases .................................................... 3
   2.1. Multihomed networks with provider assigned prefixes ... 4
   2.2. Degree of traffic engineering .......................... 5
   2.3. Distributed filtering based on source address .......... 5
   2.4. Walled-garden Enterprise services ...................... 5
   2.5. Information Source for Neighbor Management ............ 6
3. Principle of operation ....................................... 6
   3.1. Frame of reference ..................................... 6
   3.2. Route information and equality ........................ 6
   3.3. Lookup ordering and disambiguation .................... 7
   3.4. Ordering Rationale ..................................... 7
4. Routing protocol considerations ............................. 8
   4.1. Source information ..................................... 8
   4.2. Loop-freeness considerations .......................... 8
   4.3. Recursive routing ....................................... 8
5. Applicability To Specific Situations ...................... 10
   5.1. Recursive Route Lookups ................................ 10
       5.1.1. Recursive route expansion ....................... 11
   5.2. Unicast Reverse Path Filtering ......................... 12
   5.3. Multicast Reverse Path Forwarding ..................... 12
   5.4. Testing for Connectivity Availability ................. 12
6. Interoperability ............................................. 13
   6.1. Interoperability in Distance-Vector Protocols ........ 14
   6.2. Interoperability in Link-State Protocols .............. 15
7. IANA Considerations .......................................... 15
8. Security Considerations ...................................... 15
9. Privacy Considerations ....................................... 16
10. Acknowledgements ........................................... 16
11. Change Log .................................................. 16
12. References .................................................. 17
    12.1. Normative References ................................ 17
    12.2. Informative References .............................. 17
Appendix A. Implementation Options ............................ 18
    A.1. Pre-expanded 2-step lookup without backtracking ...... 19
    A.2. Translation to Multi-FIB (Policy Routing) perspective .. 19
1. Introduction

Both IPv4 [RFC0791] and IPv6 [RFC2460] architectures specify that determination of the outgoing next-hop for packet forwarding is based solely on the destination address contained in the packet header. There exists a class of network design problems which require packet forwarding to consider more than just the destination IP address (see Section 2 for examples).

At present these problems are routinely resolved by configuring special forwarding based on a local policy on routers. The policy enforces packet forwarding decision outcome based not only on the destination address but also on other fields in the packet’s IP header, most notably the source address. Such policy-based routing is conceptually similar to static routes in that it is highly static in nature and must be closely governed via the management plane (most frequently – via managing configuration by an operator). Thus policy-based routing configuration and maintenance is costly and error-prone.

Rapid expansion of IPv6 to networks where static configuration is not acceptable due to both its static nature and necessity of frequent intervention by a skilled operator requires change in the paradigm of forwarding IP packets based only on their destination address.

This document describes architecture of destination-source routing. It includes both forwarding plane and control plane considerations and requirements. Specific considerations for particular dynamic routing protocols are outside of the scope of this note and will be covered in separate documents, for example handling of a noncontiguous sub-topology in a link-state protocol.

General concepts covered by this document are equally applicable to both IPv4 and IPv6. Considering the implementation complexity of backward compatibility of destination-source routing with traditional destination-only routing, IPv4 is left outside the scope of this document.

1.1. Requirements Language

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

2. Use cases
2.1. Multihomed networks with provider assigned prefixes

There are good reasons for networks to be multihomed - benefits of doing this may include redundancy, better performance or faster access to important resources (for example, video or cloud services) local to ISPs.

However, in a range from smaller home networks to even larger enterprises, it is likely that each service provider will assign some address space (from their PA allocation) to the network.

![Diagram of multihomed network]

In this situation, providers are expected to perform ingress filtering according to BCP 38 [RFC2827]. This means only packets with a source address from the prefix that the provider assigned will be accepted. In addition, the assigned prefix can usually not be expected to remain the same.

Conventionally, NAT or policy routing would be used to produce correct behaviour. These are not desirable solutions: NAT66 breaks end-to-end connectivity (and may restrict concurrent use of parallel paths.) Policy routing requires a sufficiently skilled operator to manually manage these policies.

By assigning addresses from multiple prefixes each to end host (as a policy routing solution could do), the choice of uplink is left to host, including the option to choose multiple at once. Destination-source routing provides the necessary behaviour for routers (e.g. R1 and R2 in above example) to forward packets to the appropriate exit. It does so without requiring the manual configuration maintenance that policy routing would entail.
For a general introduction and aspects of interfacing routers to hosts, refer to [RFC8043].

2.2. Degree of traffic engineering

Consider enterprise consisting of a headquarter (HQ) and branch offices. A branch office is connected to the enterprise HQ network via 2 links. For performance or security reasons it is desired to route corporate traffic via one link and Internet traffic via another link. In direction branch -> HQ the problem is easily solvable by having the default route pointing to the Internet link and HQ routes pointing to another link. But destination routing does not provide an easy way to achieve traffic separation in direction HQ -> branch because destination is the same (branch network).

Source-destination routing provides an easy way to sort traffic going to the branch based on its source address.

2.3. Distributed filtering based on source address

A network has untrusted zone and secure one (and both zones comprise many links and routers). Computers from the secure zone need to be able to communicate with some selected hosts in the untrusted zone. The secure zone is protected by a firewall. The firewall is configured to check that packets arriving from the untrusted zone have destination address in the range of secure zone and source address of trusted hosts in the untrusted zone. This works but leaves the firewall open to DDOS attack from outside.

If routers in the untrusted zone are configured with destination-source routing (and, possibly, unicast RPF check) and receive via dynamic routing protocol routes <destination: secure zone; source: trusted host in the untrusted zone> then DDOS attack is dropped by routers on the edge of destination-source routing area. DDOS attack does not even reach the firewall whose resources are freed to deal with Deep Packet Inspection. On the other hand, security policy is managed in a single point - on a router injecting relevant destination-source routes into the dynamic routing protocol.

2.4. Walled-garden Enterprise services

Apart from transferring from multihomed personal networks to multihomed PA enterprise setups without any changes, destination-source routing can also be used to correctly route services that assign their own prefixes to customers using the particular service. This is distinct from internet connectivity only in that it does not provide a default route. Applying destination-source routing, the entire routing domain is aware of the specific constraints of the
routes involved.

Additionally, if the walled-garden’s destination prefix is advertised as blackhole route, this ensures that communication with the service will only be routed using the specific D/S route, never leaking onto unintended paths like a default route.

This is very similar to firewall/filtering functionality, except the feature is distributed onto routers.

2.5. Information Source for Neighbor Management

Having information on source address restrictions for routes distributed, routers can rely on this additional information to improve their behaviour towards hosts connected to them. This specifically includes IPv6 Router Advertisements, which is described in [RFC8028] and [I-D.linkova-v6ops-conditional-ras].

3. Principle of operation

3.1. Frame of reference

The principles described here are define on a functional level what the semantics of routing information exchanged between systems is. It is neither a prescription in how to efficiently implement these semantics, nor does it preclude an implementation from providing other administrator-friendly views of the same routing information.

More specifically, forwarding plane implementations are expected to internally diverge from the lookup algorithm described below. The router as a whole MUST ultimately behave as if the steps below were followed. An internal variation providing improved performance, as well as a variation matching existing implementations with reversed order are described in Appendix A.1 and Appendix A.2, respectively.

3.2. Route information and equality

The mechanism in this document is such that a source prefix is added to all route entries. This document assumes all entries have a source prefix, with ::/0 as default value for entries installed without a specified source prefix. This need not be implemented in this particular way, however the system MUST behave exactly as if it were. In particular, a difference in behaviour between routes with a source prefix of ::/0 and routes without source prefix MUST NOT be visible.
For uniqueness considerations, the source prefix factors MUST be taken into account for comparisons. Two routes with identical information except the source prefix MAY exist and MUST be installed and matched.

### 3.3. Lookup ordering and disambiguation

When a router is making packet forwarding decision, that is consulting its routing table in order to determine next-hop to forward the packet to, it will use information from packet’s header to look up best matching route from the routing table. This section describes lookup into the destination-source routing table.

For longest-match lookups, the source prefix is matched after the destination prefix. This is to say, first the longest matching destination prefix is found, then the table is searched for the route with the longest source prefix match, while only considering routes with exactly the destination prefix previously found. If and only if no such route exists (because none of the source prefixes match), the lookup moves to the next less specific destination prefix.

A router MUST continue to a less specific destination prefix if no route matches on the source prefix. It MUST NOT terminate lookup on such an event.

Using A < B to mean "A is more specific than B", this is represented as:

\[
A < B := \quad \text{Adst} < \text{Bdst} \\
\quad \text{||} \quad (\text{Adst} == \text{Bdst} \land \land \text{Asrc} < \text{Bsrc})
\]

### 3.4. Ordering Rationale

Ordering of searching for address match is important and reversing it would lead to semantically different behavior. This standard requires most specific match on destination address to be found before looking for match on source address.

Choosing destination to be evaluated first caters to the assumption that local networks should have full, contiguous connectivity to each other. This implies that those specific local routes always match first based on destination, and use a zero ("all sources") source prefix.
If the source prefix were to be matched first, this would result in a less specific (e.g. default) route with a source prefix to match before those local routes. In other terms, this would essentially divide local connectivity into zones based on source prefix, which is not the intention of this document.

Hence, this document describes destination-first match search.

4. Routing protocol considerations

As with the destination-only routing, destination-source routes will typically be disseminated throughout the network by dynamic routing protocols. It is expected that multiple dynamic routing protocols will be adapted to the needs of destination-source routing architecture. Specification of dynamic routing protocols is outside of scope of this document. This section lists requirements and considerations for the dynamic destination-source routing protocols.

4.1. Source information

Dynamic routing protocols will need to be able to propagate source range information together with destination prefix and other accompanying routing information. Source range information may be propagated with all destination prefixes or only some of them. Destination prefixes advertised without associated source range MUST be treated as having default source range ::/0.

Dynamic routing protocols MUST be able to propagate multiple routes whose destination prefix is the same but associated source ranges are different. Such unique pairs of destination and source MUST be treated as different destination-source routes.

There is no limitation on how source range information is propagated and associated with destination prefixes. Individual protocols may choose to propagate source range together with a destination prefix in the form of prefix, in the form of index to list of known source ranges or in any other form allowing receiver to reconstruct pair of destination prefix and associated source range.

4.2. Loop-freeness considerations

It is expected that some existing dynamic routing protocols will be enhanced to propagate destination-source routing information. In this case the protocol may be configured to operate in a network where some, but not all, routers support destination-source routing and others are still using destination-only routing. Even if all routers within a network are capable of destination-source routing, it is very likely that on edges of the network they will have to
forward packets to routers doing destination-only routing.

Since a router implementing destination-source routing can have additional, more granular routes than one that doesn’t implement it, persistent loops can form between these systems.

Thus specifications of destination-source routing protocols (either newly defined protocols or enhancements to already existing one) MUST take provisions to guarantee loop-free operations.

There are 3 possible approaches to avoid looping condition:

1. Guarantee that next-hop gateway of a destination-source route supports destination-source routing, for example calculate an alternate topology including only routers that support destination-source routing architecture

2. If next-hop gateway is not aware of destination-source routing then a destination-source path can lead to it only if next-hop router is ‘closer’ to the destination in terms of protocol’s routing metric; important particular case of the rule is if destination-only routing is pointing to the same next-hop gateway

3. Discard the packet (i.e. treat destination-source route as unreachable)

In many practical cases routing information on the edges of destination-source routing domain will be provided by an operator via configuration. Dynamic routing protocol will only disseminate this trusted external routing information. For example, returning to the use case of multihomed Home network (Section 2.1), both routers R1 and R2 will have default static routes pointing to ISPs.

Above considerations require a knowledge of the next-hop router’s capabilities. For routing protocols based on hop-by-hop flooding (RIP [RFC2080], BGP [RFC4271]), knowing the peer’s capabilities is sufficient. Information about if peer supports destination-source routing can either be negotiated explicitly or simply be deduced from the fact that systems would propagate destination-source routing information only if they understand it. Protocols building a link-state database (OSPFv3 [RFC5340], IS-IS [RFC5308]) have the additional opportunity to calculate alternate paths based on knowledge of the entire domain but cannot assume that routers understand destination-source routing information only because they participated in its flooding. Such protocols MUST explicitly advertise support for the destination-source routing.
4.3. Recursive routing

Dynamic routing protocols may propagate routing information in a recursive way. Examples of such recursion is forwarding address in OSPFv3 [RFC5340] AS-External-LSAs and NEXT_HOP attribute in BGP [RFC4271] NLRI.

Dynamic routing protocol supporting recursive routes MUST specify how this recursive routing information is interpreted in the context of destination-source routing as part of standardizing destination-source routing extensions for the protocol. Section 5.1 lists several possible strategies protocols can choose from.

5. Applicability To Specific Situations

This section discusses how destination-source routing is used together with some common networking techniques dependent on routes in the routing table.

5.1. Recursive Route Lookups

Recursive routes provide indirect path information where instead of supplying the next-hop directly they specify that next-hop information must be taken from another route in the same routing table. It is said that one route ‘recurses’ via another route which is ‘resolving’ recursion. Recursive routes may either be carried by dynamic routing protocols or provided via configuration as recursive static routes.

Recursive destination-source routes have additional complication in how source address range should be considered while finding destination-source route to resolve recursion.

There are several possible approaches:

1. Ignore destination-source routes, resolve recursion only via destination-only routes (i.e. routes with source range ::/0)

2. Require that both the recursive and resolving routes have the same source range associated with them; this requirement may be too restrictive to be useful in many cases

3. Require that source range associated with recursive route is a subset of source range associated with route resolving recursion (i.e. source range of the resolving route is less specific superset of recursive route’s source range)
4. Create multiple instances of the route whose nexthop is being
resolved with different source prefixes; this option is further
elaborated in Section 5.1.1

When recursive routing information is propagated in a dynamic routing
protocol, it is up to the protocol specification to select and
standardize appropriate scheme of recursive resolution.

Recursive resolution of configured static routes is local to router
where recursive static routes were configured, thus behavior is
implementation’s choice. Implementations SHOULD provide option (3)
from the above list as their default method of recursive static route
resolution. This is both to guarantee that destination-only
recursive static routes do not change their behavior when router’s
software is upgraded to support destination-source routing and at the
same time make destination-source recursive routes useful.

5.1.1. Recursive route expansion

When doing recursive nexthop resolution, the route that is being
resolved is installed in potentially multiple copies, inheriting all
possible more-specific routes that match the nexthop as destination.
The algorithm to do this is:

1. form the set of attributes for lookup by using the (unresolved,
   recursive) nexthop as destination (with full host prefix length,
   i.e. /128), copy all other attributes from the original route

2. find all routes that overlap with this set of attributes
   (including both more-specific and less-specific routes)

3. order the result from most to less specific

4. for each route, install a route using the original route’s
destination and the "logical and" overlap of each extra match
attribute with same attribute from the set. Copy nexthop data
from the route under iteration. Then, reduce the set of extra
attributes by what was covered by the route just installed
("logical AND NOT").

Example recursive route resolution
route to be resolved:
2001:db8:1234::/48, source 2001:db8:3456::/48,
  recursive nexthop via 2001:db8:abcd::1

routes considered for recursive nexthop:
 ::/0, current, via fe80::1
2001:db8:abcd::/48, current, via fe80::2
2001:db8:abcd::/48, source 2001:db8:3456:3::/64, via fe80::3
2001:db8:abcd::1/128, source 2001:db8:3456:4::/64, via fe80::4

recursive resolution result:
2001:db8:1234::/48, source 2001:db8:3456::/48, via fe80::2
2001:db8:1234::/48, source 2001:db8:3456:3::/64, via fe80::3
2001:db8:1234::/48, source 2001:db8:3456:4::/64, via fe80::4

5.2. Unicast Reverse Path Filtering

Unicast reverse path filtering MUST use dst-src routes analog to its usage of destination-only routes. However, the system MAY match either only incoming source against routes' destinations, or it MAY match source and destination against routes' destination and source. It MUST NOT ignore dst-src routes on uRPF checks.

5.3. Multicast Reverse Path Forwarding

Multicast Reverse Path Lookups are used to find paths towards the (known) sender of multicast packets. Since the destination of these packets is the multicast group, it cannot be matched against the source part of a dst-src route. Therefore, dst-src routes MUST be ignored for Multicast RPF lookups.

5.4. Testing for Connectivity Availability

There are situations where systems' behaviour depends on the fact whether "connectivity" is available in a broad sense. These systems may have previously tested for the existence of a default route in the routing table.

Since the default route may now be qualified with a source prefix, this test can fail. If no additional information is available to qualify this test, systems SHOULD test for the existence of any default route instead, e.g. include routes with default destination but non-default source prefix.
However, if the test can be associated with a source address or source prefix, this data SHOULD be used in looking up a default route. Depending on the application, it MAY also be useful to - possibly additionally - consider "connectivity" to be available if any route exists where the route’s source prefix covers the prefix or address under consideration, allowing arbitrary destination prefixes.

Note though that this approach to routing SHOULD NOT be used to infer a list of source prefixes in an enumerative manner, or even to guess domain information. Specifically, if an operator uses more specific source prefixes to refine their routing, the inferred information will provide bogus extraneous output. This is distinct from the connectivity tests mentioned above in that those actually inquire the routing system, unlike domain information or enumeration, which is higher-layer application information.

6. Interoperability

As pointed out in Section 4.2 traffic may permanently loop between routers forwarding packets based only on their destination IP address and routers using both source and destination addresses for forwarding decision.

In networks where the same dynamic routing protocol is being used to propagate routing information between both types of systems the protocol may address some or all traffic looping problems. Recommendations to protocol designers are discussed in Section 4.2.

When routing information is coming from outside of the routing protocol (for example, being provided by operator in the form of static routes or network protocols not aware of destination-source routing paradigm) it may not be possible for the router to ascertain loop-free properties of such routing information. In these cases consistent (and loop-free) packet forwarding is woven into network topology and must be taken into consideration at design time.

It is possible to design network with mixed deployment of routers supporting and not supporting destination-source routing. Thus gradual enablement of destination-source routing in existing networks is also possible but has to be carefully planned and evaluated for each network design individually.

Generally, destination-source routing will not cause traffic loops when disjoint ‘islands’ of destination-source routing do not exchange destination-source routing information. One particular case of this rule is a network which contains single contiguous ‘island’ of routers aware of destination-source routing. Example SOHO network from Section 2.1 which demonstrates this design approach:
Figure 2: Example of multihomed small network with partial deployment of destination-source routing

6.1. Interoperability in Distance-Vector Protocols

Distance-Vector routing protocols (BGP, RIPng, BABEL), operating on a hop-by-hop basis, can address interoperability and migration concerns on that level. With routing information being flooded in the reverse direction of traffic being forwarded using that information, a hop that floods is the same hop that forwards.

This makes dealing with destination/source-unaware routers easy if destination/source routes are made to be ignored by such unaware routers, and flooding of such routes is inhibited.

If D/S routes are discarded by non-D/S routers, D/S routers will not receive non-working routes and can select from other available working D/S routes.

Note that for this to work, non-D/S routers MUST NOT flood D/S routing information. This can be achieved in 2 ways:

1. Using some preexisting encoding to signal non-D/S routers to not flood these particular routes

2. Ignoring flooded D/S information on D/S routers by having them detect that they received it from a non-D/S router (e.g. using some capability signalling to identify non-D/S routers.) This handling likely needs to be performed on a level of same-link neighborships.
Also note that the considerations in this section only apply if data path and flooding path are congruent.

6.2. Interoperability in Link-State Protocols

For Link-State routing protocols (OSPF, IS-IS), there is no relation between route flooding and forwarding. Instead, forwarding decisions are based on shortest-path calculation on top of the received topology information.

For a D/S router to avoid loops, there are again two choices available:

1. Detect that forwarding for a D/S route transits over a non-D/S router and convert the route into a blackhole route to replace looping with blackholing. This obviously impacts connectivity.

2. Perform separate SPF calculations using only the subset of D/S-capable routers; thus D/S routers can forward D/S-routed packets as long as they stay in contiguous islands.

The latter approach is facilitated by Multi-Topology extensions to the respective protocols. These extensions provide a way to both isolate D/S routing information and perform the separate SPF calculation. Note that it is not necessary to use multiple topologies for distinct source prefixes; only a single additional topology encompassing all D/S-capable routers is sufficient.

7. IANA Considerations

This document makes no requests to IANA.

8. Security Considerations

Systems operating under the principles of this document can have routes that are more specific than the previously most specific, i.e. host routes. This can be a security concern if an operator was relying on the impossibility of hijacking such a route.

While destination-source routing could be used as part of a security solution, it is not really intended for the purpose. The approach limits routing, in the sense that it routes traffic to an appropriate egress, or gives a way to prevent communication between systems not included in a destination-source route, and in that sense could be considered similar to an access list that is managed by and scales with routing.
9. Privacy Considerations

If a host’s addresses are known, injecting a dst-src route allows isolation of traffic from that host, which may compromise privacy. However, this requires access to the routing system. As with similar problems with the destination only, defending against it is left to general mechanisms protecting the routing infrastructure.

10. Acknowledgements

The base underlying this document was first outlaid by Ole Troan and Lorenzo Colitti in [I-D.troan-homenet-sadr] for application in the homenet area. Significant contributions to source-specific routing as a whole came from Juliusz Chroboczek and Matthieu Boutier. Matthieu has also provided a huge amount of review and editing input on this document.

This document itself is largely the result of discussions with Fred Baker and derives from [I-D.baker-ipv6-isis-dst-src-routing].

Thanks to Chris Bowers, Acee Lindem and Tony Przygienda for their input and review.

The Linux kernel is providing an implementation of the behaviour described here since even before the document was started.

11. Change Log

March 2019 [-07]: no changes

October 2017 [-06]: clarify described operation is not an implementation guide

editorial cleanups

July 2017 [-05]: clarify connectivity tests

extend use cases

editorial cleanups

May 2017 [-04]: no changes

November 2016 [-03]: added DV/LS protocol considerations

note backtracking workaround/caveat

November 2015 [-02]: added section on destination-source routing use cases
added section on alternative lookup algorithm
added section on requirement for dynamic routing protocols
dissiminating destination-source information

October 2015 [-00]: renamed to draft-ietf-rtgwg-dst-src-routing-00,
no content changes from draft-lamparter-rtgwg-dst-src-routing-01.

April 2015 [-01]: merged routing-extra-qualifiers draft, new
ordering rationale section

October 2014 [-00]: Initial Version

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,

(IPv6) Specification", RFC 2460, DOI 10.17487/RFC2460,

12.2. Informative References

[hal-00947234v1]
hal 00947234v1, 2014, https://hal-univ-diderot.archives-
ouvertes.fr/hal-00947234v1.

[I-D.baker-ipv6-isis-dst-src-routing]
Baker, F. and D. Lamparter, "IPv6 Source/Destination
Routing using IS-IS", Work in Progress, Internet-Draft,
draft-baker-ipv6-isis-dst-src-routing-07, 18 July 2017,
https://www.ietf.org/archive/id/draft-baker-ipv6-isis-
dst-src-routing-07.txt.

[I-D.linkova-v6ops-conditional-ras]
Linkova, J. and M. Stucchi, "Using Conditional Router
Advertisements for Enterprise Multihoming", Work in
Progress, Internet-Draft, draft-linkova-v6ops-conditional-
draft-linkova-v6ops-conditional-ras-01.txt.


Appendix A. Implementation Options
A.1. Pre-expanded 2-step lookup without backtracking

The backtracking behavior (specified in Section 3.3 as "A router MUST continue to a less specific destination prefix") has been shown to potentially cause a significant loss of forwarding performance since forwarding a single packet may require a large number of table lookups. (The degenerate case is 129 destination lookups in decreasing prefix length, each followed by a failing longest-match on the source prefix.)

To avoid this, implementations can install synthetic routes to achieve the same lookup result. This works as follows, to be evaluated for each unique destination prefix:

1. If there is a route \((D, S=::/0)\), end processing for \(D\).

2. Iterate upwards one level (from \(D\) if first iteration, previous \(D'\) otherwise) to a less specific destination. Call this \(D'\).

3. For all routes \((D', S')\), i.e. all source prefixes \(S'\) under that destination prefix, install a copy \((D, S')\) if and only if \(S'\) covers some source prefix that isn't covered yet. (In terms of set theory, \(S'\) cut by all existing \(S\) under \(D\) is not empty.)

4. Repeat at step 1.

The effect of this algorithm is that after performing a lookup on the destination prefix, looking up the source prefix directly yields the result that backtracking would give. This eliminates backtracking and provides constant 2 lookup cost (after exactly one destination longest-match, the source longest-match will provide the final, correct result; any no-match is a final no-match).

A.2. Translation to Multi-FIB (Policy Routing) perspective

The lookup procedure described in this document requires destination-first lookup. This is not a fit with most existing implementations of Policy Routing. While Policy Routing has no formal specification, it generally permits choosing from multiple routing tables / FIBs based on, among other things, source address. Some implementations support using more than one FIB for a single lookup, but not all do.

An implementation that can choose from multiple FIBs based on source address is capable of correct forwarding according to this document, provided that it supports enough FIBs. One FIB will be used for each unique source prefix.
For a complete description of the required translation algorithm, please refer to [hal-00947234v1]. It roughly works as follows:

After destination-source routing information has been collected, one FIB table is created for each source range including the default range ::/0. Source-destination routes then replicated into each destination-only FIB table whose associated source address range is a subset of route’s source range. Note that this rule means routes with default source range ::/0 are replicated into each FIB table.

In case when multiple routes with the same destination prefix are replicated into the same FIB table only route with the most specific source address range is installed.

For example, if destination-source routing table contains these routes:

```
+=====================+==========================+==========+
| Destination prefix  | Source range             | Next Hop |
|---------------------+--------------------------+----------|
| ::/0,               | ::/0,                    | NH1      |
|                     | ::/0,                    | NH4      |
```

Table 1

then 3 FIB tables will be created associated with source ranges ::/0, 2001:db8:3456::/48 and 2001:db8:3456:8000::/56. In this example range 2001:db8:3456:8000::/56 is a subset of less specific range 2001:db8:3456::/48. Such inclusion makes a somewhat artificial example but was intentionally selected to demonstrate hierarchy of route replication.

And content of these FIB tables will be:

FIB 1 (source range ::/0):
Table 2

FIB 2 (source range 2001:db8:3456::/48):

+-----------------+----------+
| Destination prefix | Next Hop |
+-----------------+----------+
| ::/0,C          | NH1      |
| 2001:101:5678::/48, | NH4      |

Table 3

FIB 3 (source range 2001:db8:3456:8000::/56):

+-----------------+----------+
| Destination prefix | Next Hop |
+-----------------+----------+
| ::/0,C          | NH1      |
| 2001:101:1234::/48, | NH2      |
| 2001:101:5678::/48, | NH3      |
| 2001:101:abcd::/48, | NH5      |

Table 4

During packet forwarding, lookup first matches source address against the list of address ranges associated with FIB tables to select a FIB table with the most specific source address range and then does destination-only lookup in the selected FIB table.

Authors’ Addresses
David Lamparter  
NetDEF  
04103 Leipzig  
Germany  
Email: david@opensourcerouting.org

Anton Smirnov  
Cisco Systems, Inc.  
De Kleetlaan 6a  
1831 Diegem  
Belgium  
Email: as@cisco.com

Jen Linkova  
Google  
1 Darling Island Rd  
Pyrmont NSW 209  
Australia  
Email: furry@google.com

Shu Yang  
Shenzhen University  
China  
Email: yang.shu@szu.edu.cn
BGP Blockchain

draft-mcbride-rtgwg-bgp-blockchain-01

Abstract

A variety of mechanisms have been developed and deployed over the years to secure BGP including the more recent RPKI/ROA mechanisms. Is it also possible to use a distributed ledger such as Blockchain to secure BGP? BGP provides decentralized connectivity across the Internet. Blockchain provides decentralized secure transactions in a append-only, tamper-resistant ledger. This document reviews possible opportunities of using Blockchain to secure BGP policies within a domain and across the global Internet.

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 31 December 2022.

Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Introduction ............................................ 2
   1.1. Requirements Language ................................. 2
2. A Strawman for a simple BGP Distributed Consensus System .... 3
3. Opportunities for Using DCSs for BGP ......................... 4
   3.1. Preventing fraudulent BGP origin announcements ...... 5
   3.2. Validating incoming BGP updates ....................... 5
   3.3. Providing routing policy such as QoS ................. 6
   3.4. Protecting BGP files ................................. 6
   3.5. Providing path validation ............................. 6
   3.6. Securing BGP Controllers ............................. 6
   3.7. Securing Blockchain compromised by BGP vulnerabilities . 6
4. Conclusions ............................................. 7
5. IANA Considerations ....................................... 7
6. Security Considerations ................................... 7
7. Acknowledgement ........................................ 7
8. Normative References .................................... 7
Authors' Addresses ........................................ 8

1. Introduction

There have been many proposed solutions to help secure the Border Gateway Protocol (BGP) [RFC4271] including securing TCP, CoPP, IPSec, Secure BGP, Route Origination Validation (ROV), BGPSec along with many variations. Could we also use Distributed Consensus Systems (DCS) such as Blockchain to secure BGP? This document provides a review of how such DCSs could be used to secure BGP particularly as supplements to existing solutions. Many of the proposals can be extended to any routing protocol but the focus here is with BGP. The potential attractiveness of adding DCS capabilities to BGP is that it adds additional security without changes to the BGP protocol. Blockchain for BGP proposals are out of band to BGP, similar to RPKI, and not suggesting new encodings. This analysis does not consider external factors such as the energy demands of deploying such solutions.

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].
2. A Strawman for a simple BGP Distributed Consensus System

Smart contracts are programs (state machines), executed within a DCS, that run when predetermined conditions are met. These contracts are executed automatically without an intermediary’s involvement. Smart contracts may be used in financial, real estate, etc environments to automatically trigger predefined agreements between parties. A DCS implements a smart contract in the form of a distributed state machine, i.e., actions over a pool of information, where distributed DCS nodes maintain the evolving state information over time, utilizing proof techniques, such as proof-of-work, proof-of-stake, and others, to ensure consensus over the latest valid information pool (and thereby the latest state of the smart contract). In popular Blockchain systems, this information pool is represented by the longest blockchain that can be retrieved from the system by a client, i.e., representing the current consensus among the DCS nodes being queried by the client.

With this in mind, we can now describe a simple BGP DCS as one consisting of N miners, which implement the distributed consensus for a desired smart contract, utilizing a suitable proof technique for the consensus. A DCS may implement more than one smart contract, representing, e.g., different BGP capabilities as outlined later in Section 3.

In addition, there are M clients inserting transactions into the system. Those transactions relate to the desired smart contract or may be retrievals of the latest valid consensus information.

Clients and miners may be different entities or they may the same, whereby in the latter case M=N.

The figure below outlines a simple BGP DCS architecture, with BGPs providing clients to the DCS system.
In our context of BGP, we can see actions over BGP information, such as BGP origins, routing policies or others, as smart contracts over which distributed consensus needs to be achieved; Section 3 elaborates on those examples. Through using such smart contracts (over BGP information), a DCS for BGP would avoiding BGP human configuration errors or hijacks as common threats for BGP, instead storing transaction information in the DCS where the consensus here represents the latest valid BGP information.

In terms of trust assumptions, a DCS for BGP may require authentication to prevent fraudulent DCS transactions, such as fraudulent BGP announcements being made. For this, the existing RPKI system could be used to authorize any client before sending suitable smart contract transactions into the DCS. If not using RPKI, the DCS would need to check a separate IRR prefix/AS database, if one were to exist, in order to validate incoming transactions on the main DCS before executing them; such separate IRR database could be realized as a DCS itself. Furthermore, ROA entries could be added to the DCS as secure transactions and those transactions would be relied upon by route validators as authoritative. Perhaps DCS validation information could be added as a new ROA field.

In terms of openness of the system, a permissioned system would restrict both clients and miners to, e.g., AS owners, through suitable verification steps upon joining the DCS. A permissionless realisation, on the other hand, could more widely distribute the BGP origin information, still relying on the detection of fraudulent announcements through the above steps before executing a transaction.

A key requirement for realizing a suitable DCS for BGP is the latency requirement for achieving consensus, i.e., retrieving the latest valid information from the DCS. This requirement will need reflection in choosing the appropriate proof technique for consensus.

In the next section, we list several opportunities for using DCS in BGP by expressing those opportunities in smart contract language, i.e., allowing for being formulated as a distributed state machine with a distributed information pool representing the latest valid state of the system.

3. Opportunities for Using DCSs for BGP

There are various ways DCSs could be used in the context of BGP that we will explore in this section, keeping in mind the questions of the previous section.
3.1. Preventing fraudulent BGP origin announcements

BGP origin information is at the heart of BGP to ensure reachability in the global Internet, while preventing any fraudulent announcement of a BGP origin is an additional security aspect in providing this global reachability.

Announcements (of BGP origins) here represent smart contracts in a DCS, amending a distributed state (the BGP routing table), while securing those transactions prevents fraudulently doing so.

For anomaly detection purposes, we could further secure BGP origin information by comparing what’s in a BGP blockchain table against what’s in the BGP table or the forwarding table. Additional reliance upon BGP blockchain table could potentially help prevent high frequency updates from causing routing disruptions.

3.2. Validating incoming BGP updates

This is very similar to the previous aspect whereas BGP origin may not just be announced but updated, represented through a different state machine to manipulate the distributed BGP information in the DCS.

And according to RIPE labs, BGP route updates tend to converge globally in a few minutes. The propagation of newly announced prefixes happens almost instantaneously, reaching 50% visibility in under 10 seconds. Prefix withdrawals take longer to converge and generate nearly 4 times more BGP traffic, with the visibility dropping below 10% after approximately 2 minutes.

Although a DCS will likely not help with BGP updates, withdrawals may be completed faster than in existing BGP systems.

Furthermore, networking innovations that link DCS operations, like its ledger diffusion, more directly to emerging network capabilities, as suggested in [IIC_whitepaper], may improve the DCS’ transaction completion latency and thereby provide a suitable alternative even for update operations. This provides an opportunity for more research and testing.

3.3. Providing routing policy such as QoS

In addition to the prefix to AS match information being stored in the DCS, the routing policy of those routes could also be stored as part of the DCS information. As long as the policy was correctly added to the chain, the path policies cannot be altered except by those authenticated to do so.
3.4. Protecting BGP files

The DCS information could also be used to store configuration files within an AS in order to prevent malicious config tampering and to prevent misconfiguration.

This protection could be provided within a private, i.e., permissioned, DCS where only authorized users have access to the DCS data. This could also be used within a trusted external peering environment to build a distributed database of BGP files such as communities for use between BGP neighbors. Peers can use the DCS data to understand the necessary peering relationship and act on the communities in a consistent manner.

3.5. Providing path validation

BGP stores multiple paths to a destination in the BGP table. The BGP table contains all of the routes from all of the neighbors. Only the best route gets installed in the routing table. To help further secure the BGP table, all of those routes/paths could be installed in a DCS. Some mechanism could be used to validate these routes/paths, that reside in the DCS, prior to one being selected as the path in the routing table. This could also be extended to provide proof of transit across certain expected paths.

3.6. Securing BGP Controllers

BGP-LS is used to provide BGP topology information to a Controller. That topology information could be added to a DCS to ensure that the topology data is not compromised. PCEP, or other protocol, could be used by a controller to validate any update of a BGP forwarding table using this same (or separate) DCS. The latest forwarding rules would be maintained in a DCS, which is built using BGP-LS data and authorized users as an input. Without the proper credentials it would be very difficult to update the forwarding rules in the DCS and a record would be kept with all update attempts.

Furthermore, the DCS could be permissioned, thereby restricting the nodes holding as well as accessing information to trusted members of the community.

3.7. Securing Blockchain compromised by BGP vulnerabilities

The attractiveness of DCS applications, such as Bitcoin and Ethereum, are that they are highly decentralized and more resistant to attack. This has opened the way for securing monetary transactions using cryptocurrencies and their underlying blockchain technology.
Blockchains mining power, however, is centralized with mining pools concentrating within certain regions and Autonomous Systems. This also creates a more centralized routing situation which could become vulnerable to BGP vulnerabilities where IP addresses of the mining pools are hijacked. Therefore helping to further secure BGP will help to secure blockchain’s centralized mining pools, creating a circular dependency where the use of blockchains in BGP will in turn secure blockchains themselves.

4. Conclusions

This document discusses the use of distributed consensus system (DCS) techniques to complement and further secure BGP overall.

Although no specific recommendation on solutions is made, this document aims at providing first insights to think more broadly on a DCS-based infrastructure that may further enhance the capabilities of BGP as a key protocol for the Internet.

5. IANA Considerations

N/A

6. Security Considerations

There could be new blockchain related attacks that BGP would experience if blockchain were to be added into BGP’s policy system. These attacks include trying to replace the trusted chain with a fraudulent chain. We will explore some of those here or in a new draft.

7. Acknowledgement

8. Normative References

[IIC_whitepaper]


Authors’ Addresses

Mike McBride
Futurewei
Email: michael.mcbride@futurewei.com

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

David Guyman
Huawei
Email: david.guzman@huawei.com
Abstract

The requirements of computing interconnection are increasingly attracting the attention of service providers. They have been thinking about how to leverage their network advantages to provide integrated networking and computing services. This document describes some scenarios of using SRv6 based network technology which can partially meet the service requirement of computing interconnection.

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 8 January 2023.

Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

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

With the advent of new technology such as cloud computing, big data, artificial intelligence, etc., the demand for computing resource is continuously increasing. More and more data centers, intelligent computing centers, and supercomputing centers have been built to meet the growing demand of computing resource. Usually, these computing centers are centralized (e.g. central cloud). Especially, in some emerging industries, such as self-driving, cloud AR/VR, telemedicine, etc., there are not only requirements for computing resource, but also for quick delivery and guaranteed quality. These requirements are usually related to network factors, such as delay, bandwidth, and jitter, etc. These services can be deployed not only on the central cloud, but also on distributed edge nodes.

In order to coordinate computing resource at different levels (center, edge and end) uniformly, and to meet user’s requirements for computing power and network, a new type of network is proposed which can combine computing and network information and provide optimal allocation, association and scheduling of computing, storage and network resources. We call it computing interconnection network in this document. The computing interconnection network is a converged architecture with computing and network.

The computing interconnection network has attracted the attention of many service providers. Lots of service providers have proposed their own concepts of computing interconnection network, and have also released relevant technical white papers. Different computing resources are interconnected through the network. The goal of computing interconnection network is to achieve "ubiquitous computing resource, computing network symbiosis, intelligent orchestration, and integrated
services", and gradually develop into an infrastructure-level service that can be "once connected, use anywhere" like water and electricity. These goals present some challenges to current network architecture.

Segment Routing Architecture [RFC8402] proposes a network paradigm based on a source routing mechanism. The segment routing network has the remarkable characteristics of simplifying the control plane and network state. In addition, segment routing can program the network functions that need to be performed along the way, so that the packets can be transmitted and processed in the desired way. Segment routing can be applied to IPv6 network, which is called SRv6. In addition to inheriting the advantages of source routing, SRv6 has many other advantages. Firstly, IPv6 can provide more addresses to meet the needs of the Internet of Things. Secondly, SRv6 has three levels of programmability that are extremely scalable. Finally, SRv6 also supports in-suit OAM(IOAM), service chain, slicing and other features. SRv6 is the trend in the evolution of IP networks to intelligent IP networks.

The network for computing interconnection has attracted the attention of many service providers. They also have deployed new bearer network with SRv6 to provide better connection service. Based on the flexible scalability, programmability, simplicity and other advantages, SRv6 can meet some requirements of computing interconnection network. This document introduces some usage scenarios of SRv6 based computing interconnection network.

2. Usage Scenarios of SRv6 based computing interconnection network

2.1 SRv6 based computing interconnection network architecture

The following figure shows a typical architecture of SRv6 based computing interconnection network. There are two layers here, one is the infrastructure layer and the other is the control layer.
Figure 1: SRv6 based computing interconnection network architecture

*Infrastructure layer:
Edge: The network edge device of computing interconnection. In this document, Edge is both the edge device for computing interconnection and the endpoint of the SRv6 path.

Computing Resource: The computing resources connected to computing interconnection Edge. It can be cloud, edge or terminal and so on.

Client: Clients requesting computing services.

*Control layer:
Computing and Network Controller: Computing resource scheduling, orchestration and network policy distribution, in this document, CNC is used to represent it.
2.2 Path scheduling

When a computing request comes, it is necessary to decide which remote computing node is available to provide the service. Both computing power and network need to be considered for the decision. After the computing node is determined, a SRv6 path fulfilling the SLA requirement can be established to steer the packet to the destination, i.e. the remote computing node. SRv6 is based on source routing mechanism, which can compose the path information at the ingress of the network. The path information is encapsulated in the packet, and identified by a list of SIDs. Then, the packet only need to process the outermost SID downstream. The downstream nodes on the forwarding path can be stateless. SRv6 paths could be established according to default metrics (e.g. cost) or user’s policy. These paths can be in loose or strict manner.

2.3 Resource isolation

The different services for computing interconnection will be implemented on the same physical network. Some are sensitive to network delay, such as AR/VR. Some are sensitive to packet loss, such as storage services. Therefore, different services share the same physical network, but they need to be isolated from each other. This requires the network with slicing capabilities. Each slice is a logical network. Different slices can provide the services with different SLA requirements which are isolated from each other. SRv6’s programmability and protocol simplification could provide slicing capabilities. Using the programmability of SRv6, network devices can assign a specific SID and reserve hardware resources for each slice. The device identifies the network slice based on the specific SID, and steers the packet according to the topology and resources defined by the slice. Then the packets with different SLA requirements can be forwarded in different slices to meet the requirements of business isolation.

2.4 Multi segment path orchestration

As described in chapter 1, the computing interconnection network is a converged architecture with computing and network. The computing resources is interconnected through the wide area network. The network may be hierarchical, for example, including access, metro, backbone. For each computing request, CNC needs to learn the state of the network and computing resource comprehensively and make decisions according to the user’s constraint. The final selected computing node may cross multiple autonomous domains. Users may want to obtain a link with low delay, high bandwidth, or high reliability. Therefore, it is necessary to consider how to obtain a path that meets SLA when spanning multiple autonomous domains.
The SRv6’s BSID realizes the opening of network capabilities. Specifically, the SRv6 network identifies some paths which have specific SLA metrics with a SID, such as low latency. The SID is called BSID, which could be opened to CNC. CNC can select the appropriate BSID according to the user’s requirements for network. The BSID hides the complex path information, and only one SID is presented externally. The path represented by the BSID can be a complete path or a certain segment of a complete path. Using SRv6’s BSID, underlay and overlay can be combined, and multiple domains can also be connected. SRv6 based computing interconnection network can orchestrate network paths more conveniently and concisely.

2.5 Multi Service orchestration

In some scenarios, the computing service may need to pass through multiple computing service nodes. For example, in order to meet the requirements of user service’s security and stability, when data packets are transmitted in the network, they often need to pass through various service nodes in sequence, such as firewalls, IPS, and application accelerators. This can be achieved through the SRv6 service function chain (SFC for short). SRv6 SFC is realized through the programmability of SRv6. SFC uses specific SIDs to represent the value-added services. The CNC can encode the value-added service functions from the service request in the network path segment list by SIDs, and forward and process the value-added service functions along the path.

2.6 Application-aware networking

Traditionally, applications and networks are separated, and the network can only identify applications by means of five tuples. This method has a coarser granularity and cannot understand the real needs of applications for computing resource and networks. In computing interconnection network, in order to provide efficient and quality guaranteed computing services, the edge of the network is required to identify different applications and their needs through incoming packets, so as to provide different SLA services. Application-aware networking for IPv6/SRv6 (APN6 for short) can meet this demand. APN6 can carry the application identification and requirements for network and computing using IPv6 extend head. Then the network edge can perceive these applications and corresponding requirements, so as to steer the packet to the appropriate SRv6 path.

2.7 Operations, Administration and Maintenance

As an infrastructure that can serve various industry customers or individual users, the operation, administration and maintenance of computing interconnection network is very important. The network usually changes more frequently. When network quality deteriorates,
computing interconnection network needs to respond quickly and provide a better path. Therefore, real-time monitoring of the current network state is required, which can be used as the basis for more accurate and reasonable scheduling decisions and guarantee the SLA requirements. SRv6 supports in-situ OAM(IOAM), which can detect the network quality in real-time and accurate way. Based on the real-time network status, SRv6 based network can better serve computing scheduling and network SLA guarantee.

3. Collaboration between computing and network

The core concept of computing interconnection network is collaboration between computing and network. Computing and network are no longer isolated entities, and they need to cooperate with each other. Computing resources and network resources need to be managed and allocated from a global perspective. The reference architecture given in figure 1 is a possible implementation. In this architecture the centralized CNC can realize the integrated orchestration, control and management of the computing and network. The integrated orchestration of computing and network which is aimed at the diversified and customized requirements of computing and network convergence, could design product and service models based on the flexible combination of the atomic capability of the computing resource and network, and realize the unified orchestration, deployment and guarantee of computing and network services. The collaborative orchestration and scheduling of computing and network provides a new network paradigm to accelerate the digital transformation of society.

4. Best practices

Based on the above-mentioned important role of SRv6 in computing interconnection network, as a typical practice, China Mobile has built the infrastructure of SRv6 based DCI network and smart SD-WAN. The SRv6-based DCI network can uniformly access various computing resources and provide computing services. Smart SD-WAN is a new generation of SD-WAN that integrates overlay and underlay networks. SRv6 based DCI network and smart SD-WAN enhance coordination ability between computing and network resource. It enable full connectivity of end, edge, cloud and network, and combine user’s intent to achieve collaborative scheduling among application, computing and network, and improve service quality assurance capabilities, to realize end-to-end, differentiated, deterministic, and value-added computing network services.

The following is a specific application example. Considering a CDN scheduling system, CDN applications can be regarded as computing services required by users. In the traditional scheduling mode, scheduling system usually allocate CDN node according to the user’s geographic location. This will lead to a large number of users in a...
hotspot area being assigned to the same CDN node, so that some CDN nodes are busy, while others are very idle. This results in low resource utilization, and service quality cannot be guaranteed. In the computing interconnection network, CNC can manage computing resource and network resources at the same time. It can assign users in the same hotspot area to different CDN nodes according to the nodes computing load obtained in real-time. Corresponding SRv6 paths are established for steering different users’s packet to different CDN nodes. Through coordinating computing and network, the problem of unbalanced resource allocation can be solved and user service experience can be improved.

/-----------------------------------/
/                                   /
/  Computing and Network Controlle  /
/             (CNC)                 /
/--------:-------:--------:---------/
/       :        :                  /
/       :        :                  /
/............................:       :..................
/+--:----+                              :                          :
/+-------+ |                              :                          :
|CDN    | |                              :                          :
|Node 1 |-+    /-------------------------:------------------------/ :
|         / +----------+  |      Cloud     |  +----------+   /   :
|         | |                |  +----------+ /   | :
|         / +----------+  |      Specific Network|--|SDWAN CPE3|--+--+ :
|         /   *           +----------------+         *    /     | :
|         /      *                   |                  *   / +----+-:+
|         /         **SRv6 path1**|SDWAN CPE2|**SRv6 path2** /+-------+ |
|         /                               +----------+              / |CDN    | |
|         /                                |                  /  |Node 2 |-+
|         /--------------------------------+-----------------/   +-------+
|-------------|-----+-----|------------|
|---+--+      |---+--+     |---+--+     |---+---+|
|user 1|-+    |user 2|-+   |user 3|-+   |user 4|-+
|------|      |------|     |------|     |------|

Figure 2: CDN system with SRv6 based network

Specifically, as shown in figure 2, CDN Node 1 and CDN Node 2 are located at SD-WAN CPE1 and SD-WAN CPE3 respectively. There are a large number of users in the same area accessing to SD-WAN CPE2. It is assumed that CDN Node 1 is closer to this area. In the traditional way, user 1 to user 4 all access CDN Node 1. After the computing interconnection network is deployed, CNC will consider the network and computing load factors at the same time. User 1 and user 2 will access
CDN Node 1. User 3 and user 4 will access CDN Node 2. Meanwhile, two SRv6 paths is established on SD-WAN CPE2. Path 1 is to CDN Node 1 and another is to CDN Node 2. User 1 and user 2 will access through path 1, and user 3 and user 4 will access another. In this way, better resource utilization and service experience can be achieved.

5. Security Considerations
   To be done.

6. IANA Considerations
   This document does not make any IANA request.

7. Informative References


Authors’ Addresses

Xiaoqiu Zhang
China Mobile
Email: zhangxiaoqiu@chinamobile.com

Feng Yang
China Mobile
Email: yangfeng@chinamobile.com

Weiqiang Cheng
China Mobile
Email: chengweiqiang@chinamobile.com

Zhihua Fu
New H3C Technologies
Email: fuzhihua@h3c.com