RTGWG
Internet-Draft
Intended status: Informational
Expires: April 30, 2015

W. Atwood
N. Prajapati
Concordia University/CSE

October 27, 2014

A Framework for Secure Routing Protocols
draft-atwood-rtgwg-secure-rtg-00

Abstract

When tightening the security of the core routing infrastructure, two steps are necessary. The first is to secure the routing protocols' packets on the wire. The second is to ensure that the keying material for the routing protocol exchanges is distributed only to the appropriate routers. This document specifies a way of organizing the security parameters and a method for conveniently controlling those parameters using YANG and NETCONF.

Status of This Memo

This Internet-Draft is submitted to IETF 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 http://datatracker.ietf.org/drafts/current/.

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

This Internet-Draft will expire on April 30, 2015.

Table of Contents

1. Introduction

Much effort has been expended to ensure the security of end-to-end exchanges in the Internet. However, relatively little effort appears to be being expended to secure the router-to-router exchanges that define the forwarding path for the packets that make up the end-to-end exchanges.

Methods for ensuring router-to-router security have been written into the specifications of routing protocols for many years. However, the security parameters (keys, permitted neighbors, etc.) are typically installed manually on each router [RFC6862]. Because network management personnel are scarce, and updating security parameters is a labor-intensive task, if security is implemented at all, the keys are often left in place for five years or more [RFC6862], leaving ample opportunity for them to be compromised. This could lead to an intruder router pretending to be a legitimate one and capturing confidential data.

In March 2006, the Internet Architecture Board (IAB) held a workshop on the topic "Unwanted Internet Traffic". The report from that workshop is documented in [RFC4948]. Section 8.1 of that document states, "A simple risk analysis would suggest that an ideal attack target of minimal cost but maximal disruption is the core routing infrastructure". Section 8.2 calls for "[t]ightening the security of the core routing infrastructure".

One approach to achieving improved security is to automate the process of updating the security parameters. This will reduce the number of network management personnel needed and would potentially improve security for all users of the Internet. This leads us to the following requirements:

- Ensuring the authenticity and integrity of the routing protocol messages;
- Ensuring the legitimacy of the neighboring routers, by making sure that they are part of the "permitted adjacency" as explained below;
- Automation of the entire process of key and adjacency management.

The notion of "permitted adjacency" can be re-stated as providing answers to the following questions:

- Are you a legitimate member of my group? This is the question of authentication.
- Are you permitted to connect to me for the purposes of this routing protocol? This is the question of authorization.

Figure 1 shows a potential framework for discussion of secure routing.

Routing Protocol	Layer 1
Keys and Security Protocol	Layer 2
Key and SA Management	Layer 3
Configuration Management	Layer 4

Figure 1: Secure Routing Framework

Layer 1 is the routing protocol layer. The routers run routing protocols among themselves to collect and distribute topological information for the network. The routing protocols distribute the network information by "exchanging messages" with their peer routers (neighbors). Each router processes all the information received from the routing protocol peers to create and maintain the forwarding table. This forwarding table is used to decide where to forward a particular packet when it arrives.

Layer 2 represents the security mechanisms available for a routing protocol. Each routing protocol will have a number of security mechanisms available to it (including no security at all). A routing protocol needs to be assured of two things about the messages that it receives from its peer routers:

- that the peer is legitimate, and
- that the message from that peer has not been altered in transit.

The most common approach today is for a routing protocol to use a pre-shared key for authorizing its neighbors as well as for validating the message integrity. In effect, all the neighbors (running the same routing protocol) that possess this key are authorized to communicate with each other.

The configuration of keys and security associations, the choice of keys and the security mechanism used for a routing protocol depend on the key management methods at Layer 3. As discussed in Section 1, the network operators use the manual management method, which is the only solution available at this time for routing protocols. As a result, keys are seldom changed.

Layer 4 focuses on the configuration and the distribution of keys and security associations for routing protocols. At this time, this is done manually, either by visiting the router itself, or accessing it remotely through some configuration procedure. Each router manufacturer has its own approach to faciltiate this.

Within the KARP Working Group, protocols and procedures for creating shared keys for specific environments have been proposed [I-D.hartman-karp-mrkmp][I-D.mahesh-karp-rkmp][I-D.tran-karp-mrmp], under the assumption that the end points of the exchanges (the routers) are entitled to enter into the conversation, i.e., that they can prove that they are who they say they are. However, this only addresses part of the problem at Layer 3, because these documents provide no mechanism to assess or ensure that the end points are entitled to be neighbors.

In addition, requirements for an operations and management model are specified in [RFC7210].

This document addresses two issues: providing a flexible method for managing the necessary keys and security associations, and providing a way to configure a set of routers while satisfying operational constraints.

2. Routing Protocol Security

To be able to effectively manage routing protocol security, it is necessary to have a representation of the choices open to a key negotiation protocol, and to have a convenient representation of the parameters to be used in a particular security association that is being used by the security features of a routing protocol.

The representation of parameters (keys and security associations, key derivation functions) is provided by the Crypto-Key-Table specified in [RFC7210].

The parameters for a specific peer router and protocol are provided in the Routing Security Parameter Database (RSPD). The Routing Peer Authorization Database (RPAD) provides information required for peer authentication and authorization and specifies a key management protocol to be used in establishing the peer relationship.

3. RPsec Configuration

To enable convenient configuration of the RPsec databases, YANG models of these databases can be used, in conjunction with a central controller to define updates to the security configurations.

4. RPsec Databases

4.1. RSPD

The objective of the RSPD is to provide security options (choice of security protocol) for a routing protocol's security. Each entry (a choice) specifies the security parameters required to establish a security association between the peers. An authorized device may communicate with many routing protocol peers. To do so, it must agree on the security requirements of the routing protocol peer for successful communication. The peers must agree on security protocols, transforms, mode of communication along with the key required to integrity protect messages exchanged between them. This database aims to provide such information. The RSPD contains the traffic descriptors for identifying each routing protocol traffic that needs to be protected, bypassed or discarded. The RSPD, thus, is a database to specify the traffic descriptors for the routing protocol traffic, security protocols, lifetime and related parameters for securing the communication between the two devices or among a group in case of the multicast communication. This database provides partial information towards security requirements of the routing protocols. The rest of the information is provided by the CKT.

4.2. CKT

The CKT is an important database that provisions key material and associated cryptographic algorithms to protect the routing protocol messages. In RPsec, the CKT performs the role similar to the SAD in IPsec. It stores the negotiated (or manually configured) SAs for the routing protocols. In that, each RSPD entry points to an appropriate entry in the CKT. Each RSPD entry that protects the routing protocol traffic, provides a (security) protocol id and a peer id (traffic descriptor) that identify an entry in this database. The form of the protocol id and the peer id is specified in [RFC7210]. The RSPD together with CKT ensure that the key is provided to a security protocol that is used for securing the routing protocol.

4.3. RPAD

The RPAD's objective is to provide authentication information and a KMP for the routing peers. It provides authentication information necessary to assert a local device's identity and to validate the identity asserted by the peer devices. A KMP uses the information in the RPAD and the RSPD for authentication and SA negotiation, respectively. Authentication is required to ensure that the devices participating in the network infrastructure are legitimate. A legitimate device should present its identity, identity of remote peer(s) or group it wishes to communicate with, and an organization-wide acceptable credential. If the device successfully passes the peer device's scrutiny, it is authenticated to communicate with the requested peer(s) or a group in the network. The communication between the two devices must stop if the KMP fails to authenticate the peers using the information available in the RPAD database. A KMP negotiates a security association only after the authentication is successful.

5. RPsec in Detail

Detailed design of the RPsec databases. To be included in the next version of the draft.

6. Representation and Distribution of RPsec Policies

This section explains the YANG models for each RPsec database. It describes a possible way of configuring RPsec databases in the network in compiance with the IETF's policy-based network managment (PBMN) and distributed management architecture.

For management of the contents of the RPsec databases, the data fields of the RPsec databases are organized and defined in four modules:

- RPsec common types module
- RPAD module
- RSPD module
- CKT module

The material on YANG models will be included in the next version of the draft.

7. IANA Considerations

This document has no actions for IANA.

8. Acknowledgements

The original idea for the RAPD database was presented in [I-D.atwood-karp-aapm-rp].

9. Change History (RFC Editor: Delete Before Publishing)

[NOTE TO RFC EDITOR: this section for use during I-D stage only. Please remove before publishing as RFC.]

atwood-rtgwg-secure-routing-00 (original submission, based on Nitin's thesis)

- copied in some sections of the thesis that are relevant to the specification.

10. Needs Work in Next Draft (RFC Editor: Delete Before Publishing)

[NOTE TO RFC EDITOR: this section for use during I-D stage only. Please remove before publishing as RFC.]

List of stuff that still needs work

- Flesh out sections on RPsec databases and YANG models.
-
-
-
-

11. References

11.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

11.2. Informative References

[I-D.atwood-karp-aapm-rp]
william.atwood@concordia.ca, w., Somanatha, R., Hartman, S., and D. Zhang, "Authentication, Authorization and Policy Management for Routing Protocols", draft-atwood-karp-aapm-rp-00 (work in progress), July 2013.

[I-D.hartman-karp-mrkmp]
Hartman, S., Zhang, D., and G. Lebovitz, "Multicast Router Key Management Protocol (MaRK)", draft-hartman-karp-mrkmp-05 (work in progress), September 2012.

[I-D.mahesh-karp-rkmp]
Jethanandani, M., Weis, B., Patel, K., Zhang, D., Hartman, S., Chunduri, U., Tian, A., and J. Touch, "Negotiation for Keying Pairwise Routing Protocols in IKEv2", draft-mahesh-karp-rkmp-05 (work in progress), November 2013.

[I-D.tran-karp-mrmp]
Tran, P. and B. Weis, "The Use of G-IKEv2 for Multicast Router Key Management", draft-tran-karp-mrmp-02 (work in progress), October 2012.

[RFC2409] Harkins, D. and D. Carrel, "The Internet Key Exchange (IKE)", RFC 2409, November 1998.

[RFC3740] Hardjono, T. and B. Weis, "The Multicast Group Security Architecture", RFC 3740, March 2004.

[RFC4535] Harney, H., Meth, U., Colegrove, A., and G. Gross, "GSAKMP: Group Secure Association Key Management Protocol", RFC 4535, June 2006.

[RFC4948] Andersson, L., Davies, E., and L. Zhang, "Report from the IAB workshop on Unwanted Traffic March 9-10, 2006", RFC 4948, August 2007.

[RFC5374] Weis, B., Gross, G., and D. Ignjatic, "Multicast Extensions to the Security Architecture for the Internet Protocol", RFC 5374, November 2008.

[RFC5796] Atwood, W., Islam, S., and M. Siami, "Authentication and Confidentiality in Protocol Independent Multicast Sparse Mode (PIM-SM) Link-Local Messages", RFC 5796, March 2010.

[RFC5996] Kaufman, C., Hoffman, P., Nir, Y., and P. Eronen, "Internet Key Exchange Protocol Version 2 (IKEv2)", RFC 5996, September 2010.

[RFC6407] Weis, B., Rowles, S., and T. Hardjono, "The Group Domain of Interpretation", RFC 6407, October 2011.

[RFC6518] Lebovitz, G. and M. Bhatia, "Keying and Authentication for Routing Protocols (KARP) Design Guidelines", RFC 6518, February 2012.

[RFC6862] Lebovitz, G., Bhatia, M., and B. Weis, "Keying and Authentication for Routing Protocols (KARP) Overview, Threats, and Requirements", RFC 6862, March 2013.

[RFC7210] Housley, R., Polk, T., Hartman, S., and D. Zhang, "Database of Long-Lived Symmetric Cryptographic Keys", RFC 7210, April 2014.

[RFC7211] Hartman, S. and D. Zhang, "Operations Model for Router Keying", RFC 7211, June 2014.

Authors' Addresses

William Atwood
Concordia University/CSE
1455 de Maisonneuve Blvd, West
Montreal, QC H3G 1M8
Canada

Phone: +1(514)848-2424 ext3046
Email: william.atwood@concordia.ca
URI: http://users.encs.concordia.ca/~bill

Nitin Prajapati
Concordia University/CSE
1455 de Maisonneuve Blvd, West
Montreal, QC H3G 1M8
Canada

Email: prajapatinitin@hotmail.com

Working Group
Internet-Draft
Intended status: Informational
Expires: March 12, 2016

U. Chunduri
J. Tantsura
Ericsson Inc.
C. Bowers
Juniper Networks
September 9, 2015

Extended procedures and considerations for evaluating Loop-Free Alternates
draft-chunduri-rtgwg-lfa-extended-procedures-03

Abstract

This document provide few clarifications and extended procedures to IP Fast Reroute using Loop-Free Alternates as defined in RFC 5286.

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 http://datatracker.ietf.org/drafts/current/.

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

This Internet-Draft will expire on March 12, 2016.

Copyright Notice

Table of Contents

1. Introduction

Loop Free Alternatives (LFAs) as defined in [RFC5286] have been widely deployed, and the operational and manageability considerations are described in great detail in [I-D.ietf-rtgwg-lfa-manageability].

This document intends to provide clarifications, additional considerations to [RFC5286], to address a few coverage and operational observations. These observations are in the area of handling Muti-homed prefixes (MHPs), IS-IS attach (ATT) bit in L1 area, links provisioned with MAX_METRIC for traffic engineering (TE) purposes and in the area of Multi Topology (MT) IGP deployments. All these are elaborated in detail in Section 2.

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].

1.2. Acronyms

AF - Address Family

ATT - IS-IS Attach Bit

ECMP - Equal Cost Multi Path

IGP - Interior Gateway Protocol

IS-IS - Intermediate System to Intermediate System

OSPF - Open Shortest Path First

MHP - Multi-homed Prefix

MT - Multi Topology

SPF - Shortest Path First PDU

2. LFA Extended Procedures

This section explains the additional considerations in various aspects as listed below to the base LFA specification [RFC5286].

2.1. Multi Homed Prefixes

LFA base specification [RFC5286] Section 6.1 recommends that a router compute the alternate next-hop for an IGP multi-homed prefix by considering alternate paths via all routers that have announced that prefix. However, it also allows for the router to simplify the multi-homed prefix calculation by assuming that the MHP is solely attached to the router that was its pre-failure optimal point of attachment, at the expense of potentially lower coverage. If an implementation chooses to simplify the multi-homed prefix calculation by assuming that the MHP is solely attached to the router that was its pre-failure optimal point of attachment, the procedure described in this memo can potentially improve coverage for equal cost multi path (ECMP) MHPs without incurring extra computational cost.

The approach as specified in [RFC5286] Section 6.1 last paragraph, is to simplify the MHP is solely attached to the router that was its pre-failure optimal point of attachment. While this is very scalable approach and simplifies computation, as [RFC5286] notes this may result in little less coverage.

This memo improves the above approach to provide loop-free alternatives without any additional cost for equal cost multi path MHPs as described through the below example network. The approach specified here MAY also applicable for handling default routes as explained in Section 2.1.1.

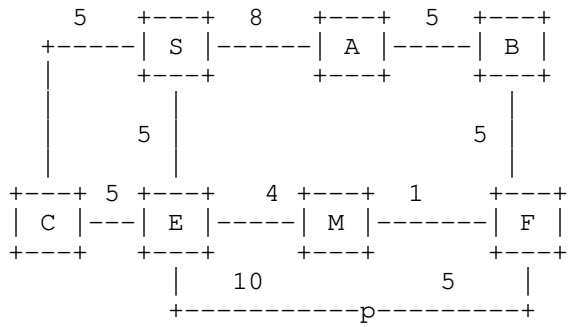

Figure 1: MHP with same ECMP Next-hop

In the above network a prefix p, is advertised from both Node E and Node F. With simplified approach taken as specified in [RFC5286] Section 6.1, prefix p will get only link protection LFA through the neighbor C while a node protection path is available through neighbor A. In this scenario, E and F both are pre-failure optimal points of attachment and share the same primary next-hop. Hence, an implementation MAY compare the kind of protection A provides to F (link-and-node protection) with the kind of protection C provides to E (link protection) and inherit the better alternative to prefix p and here it is A.

However, in the below network prefix p has an ECMP through both node E and node F with cost 20. Though it has 2 pre-failure optimal points of attachment, the primary next-hop to each pre-failure optimal point of attachment is different. In this case, prefix p shall inherit corresponding LFA to each primary next-hop calculated for the router advertising the same respectively (node E's and node F's LFA).

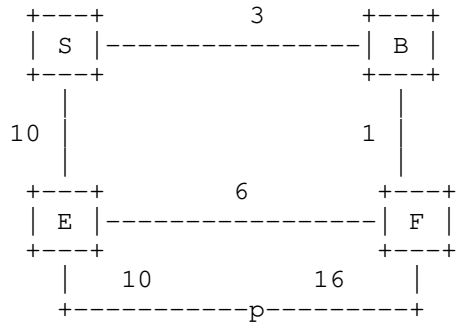

Figure 2: MHP with different ECMP Next-hops

In summary, if there are multiple pre-failure points of attachment for a MHP and primary next-hop of a MHP is same as that of the primary next-hop of the router that was pre-failure optimal point of attachment, an implementation MAY provide the better protection to MHP without incurring any additional computation cost.

2.1.1. IS-IS ATT Bit considerations

Per [RFC1195] a default route needs to be added in Level1 (L1) router to the closest reachable Level1/Level2 (L1/L2) router in the network advertising ATT (attach) bit in its LSP-0 fragment. All L1 routers in the area would do this during the decision process with the next-hop of the default route set to the adjacent router through which the closest L1/L2 router is reachable. The base LFA specification [RFC5286] does not specify any procedure for computing LFA for a default route in IS-IS L1 area. Potentially one MAY consider a default route is being advertised from the boarder L1/L2 router where ATT bit is set and can do LFA computation for the default route. But, when multiple ECMP L1/L2 routers are reachable in an L1 area corresponding best LFAs SHOULD be given for each primary next-hop associated with default route. Considerations as specified in Section 2.1 are applicable for default routes, if the default route is considered as ECMP MHP.

2.2. Links with IGP MAX_METRIC

Section 3.5 and 3.6 of [RFC5286] describes procedures for excluding nodes and links from use in alternate paths based on the maximum link metric (as defined in for IS-IS in [RFC5305] or as defined in [RFC3137] for OSPF). If these procedures are strictly followed, there are situations, as described below, where the only potential alternate available which satisfies the basic loop-free condition will not be considered as alternative.

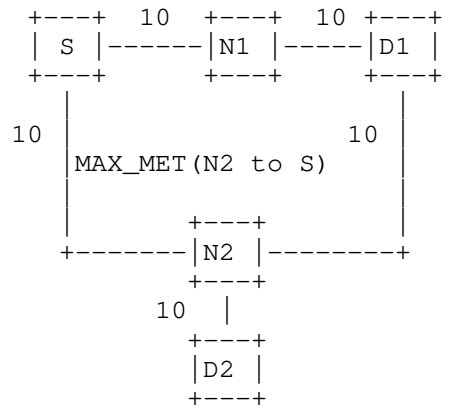

Figure 3: Link with IGP MAX_METRIC

In the simple example network, all the link costs have a cost of 10 in both directions, except for the link between S and N2. The S-N2 link has a cost of 10 in the direction from S to N2, and a cost of MAX_METRIC in the direction from N2 to S (0xffffff /2^24 - 1 for IS-IS and 0xffff for OSPF) for a specific end to end Traffic Engineering (TE) requirement of the operator. At node S, D1 is reachable through N1 with cost 20, and D2 is reachable through N2 with cost 20. Even though neighbor N2 satisfies basic loop-free condition (inequality 1 of [RFC5286]) for D1 this could be excluded as potential alternative because of the current exclusions as specified in section 3.5 and 3.6 procedure of [RFC5286]. But, as the primary traffic destined to D2 is continue to use the link and hence irrespective of the reverse metric in this case, the same link MAY be used as a potential LFA for D1.

Alternatively, reverse metric of the link MAY be configured with MAX_METRIC-1, so that the link can be used as an alternative while meeting the TE requirements.

2.3. Multi Topology Considerations

Section 6.2 and 6.3.2 of [RFC5286] state that multi-topology OSPF and ISIS are out of scope for that specification. This memo clarifies and describes the applicability.

In Multi Topology (MT) IGP deployments, for each MT ID, a separate shortest path tree (SPT) is built with topology specific adjacencies, the LFA principles laid out in [RFC5286] are actually applicable for MT IS-IS [RFC5120] LFA SPF. The primary difference in this case is, identifying the eligible-set of neighbors for each LFA computation

which is done per MT ID. The eligible-set for each MT ID is determined by the presence of IGP adjacency from Source to the neighboring node on that MT-ID apart from the administrative restrictions and other checks laid out in [RFC5286]. The same is also applicable for OSPF [RFC4915] [MT-OSPF] or different AFs in multi instance OSPFv3 [RFC5838].

However for MT IS-IS, if a default topology is used with MT-ID 0 [RFC5286] and both IPv4 [RFC5305] and IPv6 routes/AFs [RFC5308] are present, then the condition of network congruency is applicable for LFA computation as well. Network congruency here refers to, having same address families provisioned on all the links and all the nodes of the network with MT-ID 0. Here with single decision process both IPv4 and IPv6 next-hops are computed for all the prefixes in the network and similarly with one LFA computation from all eligible neighbors per [RFC5286], all potential alternatives can be computed.

3. IANA Considerations

This document defines no new namespaces and no actions for IANA.

4. Security Considerations

This document does not introduce any new security issues or any change in security considerations as noted in the LFA base specification [RFC5286].

5. Acknowledgements

Authors would like to thank Alia Atlas for detailed review of initial document and providing valuable suggestions. We also thank Bruno Decreane, Stephane Litkowski for their initial review and feedback on the document.

6. References

6.1. Normative References

[RFC1195] Callon, R., "Use of OSI IS-IS for routing in TCP/IP and dual environments", RFC 1195, DOI 10.17487/RFC1195, December 1990, <http://www.rfc-editor.org/info/rfc1195>.

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

[RFC5286] Atlas, A., Ed. and A. Zinin, Ed., "Basic Specification for IP Fast Reroute: Loop-Free Alternates", RFC 5286, DOI 10.17487/RFC5286, September 2008, <http://www.rfc-editor.org/info/rfc5286>.

6.2. Informative References

[I-D.ietf-rtgwg-lfa-manageability] Litkowski, S., Decraene, B., Filsfils, C., Raza, K., Horneffer, M., and P. Sarkar, "Operational management of Loop Free Alternates", draft-ietf-rtgwg-lfa-manageability-11 (work in progress), June 2015.

[RFC3137] Retana, A., Nguyen, L., White, R., Zinin, A., and D. McPherson, "OSPF Stub Router Advertisement", RFC 3137, DOI 10.17487/RFC3137, June 2001, <http://www.rfc-editor.org/info/rfc3137>.

[RFC4915] Psenak, P., Mirtorabi, S., Roy, A., Nguyen, L., and P. Pillay-Esnault, "Multi-Topology (MT) Routing in OSPF", RFC 4915, DOI 10.17487/RFC4915, June 2007, <http://www.rfc-editor.org/info/rfc4915>.

[RFC5120] Przygienda, T., Shen, N., and N. Sheth, "M-ISIS: Multi Topology (MT) Routing in Intermediate System to Intermediate Systems (IS-ISs)", RFC 5120, DOI 10.17487/RFC5120, February 2008, <http://www.rfc-editor.org/info/rfc5120>.

[RFC5305] Li, T. and H. Smit, "IS-IS Extensions for Traffic Engineering", RFC 5305, DOI 10.17487/RFC5305, October 2008, <http://www.rfc-editor.org/info/rfc5305>.

[RFC5308] Hopps, C., "Routing IPv6 with IS-IS", RFC 5308, DOI 10.17487/RFC5308, October 2008, <http://www.rfc-editor.org/info/rfc5308>.

[RFC5838] Lindem, A., Ed., Mirtorabi, S., Roy, A., Barnes, M., and R. Aggarwal, "Support of Address Families in OSPFv3", RFC 5838, DOI 10.17487/RFC5838, April 2010, <http://www.rfc-editor.org/info/rfc5838>.

Authors' Addresses

Uma Chunduri
Ericsson Inc.
300 Holger Way,
San Jose, California 95134
USA

Phone: 408 750-5678
Email: uma.chunduri@ericsson.com

Jeff Tantsura
Ericsson Inc.
300 Holger Way,
San Jose, California 95134
USA

Email: jeff.tantsura@ericsson.com

Chris Bowers
Juniper Networks
1194 N. Mathilda Ave.
Sunnyvale, California 94089
USA

Email: cbowers@juniper.net

NETMOD Working Group
Internet-Draft
Intended status: Standards Track
Expires: May 7, 2017

L. Lhotka
CZ.NIC
A. Lindem
Cisco Systems

November 03, 2016

A YANG Data Model for Routing Management
draft-ietf-netmod-routing-cfg-25

Abstract

This document contains a specification of three YANG modules and one submodule. Together they form the core routing data model which serves as a framework for configuring and managing a routing subsystem. It is expected that these modules will be augmented by additional YANG modules defining data models for control plane protocols, route filters and other functions. The core routing data model provides common building blocks for such extensions -- routes, routing information bases (RIB), and control plane protocols.

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 http://datatracker.ietf.org/drafts/current/.

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

This Internet-Draft will expire on May 7, 2017.

Table of Contents

1. Introduction

This document contains a specification of the following YANG modules:

- Module "ietf-routing" provides generic components of a routing data model.

- Module "ietf-ipv4-unicast-routing" augments the "ietf-routing" module with additional data specific to IPv4 unicast.

- Module "ietf-ipv6-unicast-routing" augments the "ietf-routing" module with additional data specific to IPv6 unicast. Its submodule "ietf-ipv6-router-advertisements" also augments the "ietf-interfaces" [RFC7223] and "ietf-ip" [RFC7277] modules with IPv6 router configuration variables required by [RFC4861].

These modules together define the so-called core routing data model, which is intended as a basis for future data model development covering more sophisticated routing systems. While these three modules can be directly used for simple IP devices with static routing (see Appendix B), their main purpose is to provide essential building blocks for more complicated data models involving multiple control plane protocols, multicast routing, additional address families, and advanced functions such as route filtering or policy routing. To this end, it is expected that the core routing data model will be augmented by numerous modules developed by other IETF working groups.

2. Terminology and Notation

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].

The following terms are defined in [RFC6241]:

- client,
- message,
- protocol operation,
- server.

The following terms are defined in [RFC7950]:

- action,
- augment,
- configuration data,
- container,
- container with presence,
- data model,
- data node,
- feature,
- leaf,
- list,
- mandatory node,
- module,
- schema tree,
- state data,
- RPC operation.

2.1. Glossary of New Terms

core routing data model: YANG data model comprising "ietf-routing", "ietf-ipv4-unicast-routing" and "ietf-ipv6-unicast-routing" modules.

direct route: a route to a directly connected network.

routing information base (RIB): An object containing a list of routes together with other information. See Section 5.2 for details.

system-controlled entry: An entry of a list in state data ("config false") that is created by the system independently of what has been explicitly configured. See Section 4.1 for details.

user-controlled entry: An entry of a list in state data ("config false") that is created and deleted as a direct consequence of certain configuration changes. See Section 4.1 for details.

2.2. Tree Diagrams

A simplified graphical representation of the complete data tree is presented in Appendix A, and similar diagrams of its various subtrees appear in the main text.

- Brackets "[" and "]" enclose list keys.
- Curly braces "{" and "}" contain names of optional features that make the corresponding node conditional.
- Abbreviations before data node names: "rw" means configuration (read-write), "ro" state data (read-only), "-x" RPC operations or actions, and "-n" notifications.
- Symbols after data node names: "?" means an optional node, "!" a container with presence, and "*" denotes a "list" or "leaf-list".
- Parentheses enclose choice and case nodes, and case nodes are also marked with a colon (":").
- Ellipsis ("...") stands for contents of subtrees that are not shown.

2.3. Prefixes in Data Node Names

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

Prefix	YANG module	Reference
if	ietf-interfaces	[RFC7223]
ip	ietf-ip	[RFC7277]
rt	ietf-routing	Section 7
v4ur	ietf-ipv4-unicast-routing	Section 8
v6ur	ietf-ipv6-unicast-routing	Section 9
yang	ietf-yang-types	[RFC6991]
inet	ietf-inet-types	[RFC6991]

Table 1: Prefixes and corresponding YANG modules

3. Objectives

The initial design of the core routing data model was driven by the following objectives:

- The data model should be suitable for the common address families, in particular IPv4 and IPv6, and for unicast and multicast routing, as well as Multiprotocol Label Switching (MPLS).

- A simple IP routing system, such as one that uses only static routing, should be configurable in a simple way, ideally without any need to develop additional YANG modules.

- On the other hand, the core routing framework must allow for complicated implementations involving multiple routing information bases (RIB) and multiple control plane protocols, as well as controlled redistributions of routing information.

- Device vendors will want to map the data models built on this generic framework to their proprietary data models and configuration interfaces. Therefore, the framework should be flexible enough to facilitate such a mapping and accommodate data models with different logic.

4. The Design of the Core Routing Data Model

The core routing data model consists of three YANG modules and one submodule. The first module, "ietf-routing", defines the generic components of a routing system. The other two modules, "ietf-ipv4-unicast-routing" and "ietf-ipv6-unicast-routing", augment the "ietf-routing" module with additional data nodes that are needed for IPv4 and IPv6 unicast routing, respectively. Module "ietf-ipv6-unicast-routing" has a submodule, "ietf-ipv6-router-advertisements", that augments the "ietf-interfaces" [RFC7223] and "ietf-ip" [RFC7277] modules with configuration variables for IPv6 router advertisements as required by [RFC4861]. Figures 1 and 2 show abridged views of the configuration and state data hierarchies. See Appendix A for the complete data trees.

```
+--rw routing
   +--rw router-id?
   +--rw control-plane-protocols
   |  +--rw control-plane-protocol* [type name]
   |     +--rw type
   |     +--rw name
   |     +--rw description?
   |     +--rw static-routes
   |        +--rw v6ur:ipv6
   |        |     ...
   |        +--rw v4ur:ipv4
   |              ...
   +--rw ribs
      +--rw rib* [name]
         +--rw name
         +--rw address-family?
         +--rw description?
```

Figure 1: Configuration data hierarchy.

```
+--ro routing-state
   +--ro router-id?
   +--ro interfaces
   |  +--ro interface*
   +--ro control-plane-protocols
   |  +--ro control-plane-protocol* [type name]
   |     +--ro type
   |     +--ro name
   +--ro ribs
      +--ro rib* [name]
         +--ro name
         +--ro address-family
         +--ro default-rib?
         +--ro routes
         |  +--ro route*
         |        ...
```

Figure 2: State data hierarchy.

As can be seen from Figures 1 and 2, the core routing data model introduces several generic components of a routing framework: routes, RIBs containing lists of routes, and control plane protocols. Section 5 describes these components in more detail.

4.1. System-Controlled and User-Controlled List Entries

The core routing data model defines several lists in the schema tree, such as "rib", that have to be populated with at least one entry in any properly functioning device, and additional entries may be configured by a client.

In such a list, the server creates the required item as a so-called system-controlled entry in state data, i.e., inside the "routing-state" container.

An example can be seen in Appendix D: the "/routing-state/ribs/rib" list has two system-controlled entries named "ipv4-master" and "ipv6-master".

Additional entries may be created in the configuration by a client, e.g., via the NETCONF protocol. These are so-called user-controlled entries. If the server accepts a configured user-controlled entry, then this entry also appears in the state data version of the list.

Corresponding entries in both versions of the list (in state data and configuration) have the same value of the list key.

A client may also provide supplemental configuration of system-controlled entries. To do so, the client creates a new entry in the configuration with the desired contents. In order to bind this entry to the corresponding entry in the state data list, the key of the configuration entry has to be set to the same value as the key of the state entry.

Deleting a user-controlled entry from the configuration list results in the removal of the corresponding entry in the state data list. In contrast, if a system-controlled entry is deleted from the configuration list, only the extra configuration specified in that entry is removed but the corresponding state data entry remains in the list.

5. Basic Building Blocks

This section describes the essential components of the core routing data model.

5.1. Route

Routes are basic elements of information in a routing system. The core routing data model defines only the following minimal set of route attributes:

- "destination-prefix": address prefix specifying the set of destination addresses for which the route may be used. This attribute is mandatory.

- "route-preference": an integer value (also known as administrative distance) that is used for selecting a preferred route among routes with the same destination prefix. A lower value means a more preferred route.

- "next-hop": determines the outgoing interface and/or next-hop address(es), other operation to be performed with a packet.

Routes are primarily state data that appear as entries of RIBs (Section 5.2) but they may also be found in configuration data, for example as manually configured static routes. In the latter case, configurable route attributes are generally a subset of attributes defined for RIB routes.

5.2. Routing Information Base (RIB)

Every implementation of the core routing data model manages one or more routing information bases (RIB). A RIB is a list of routes complemented with administrative data. Each RIB contains only routes

of one address family. An address family is represented by an identity derived from the "rt:address-family" base identity.

In the core routing data model, RIBs are state data represented as entries of the list "/routing-state/ribs/rib". The contents of RIBs are controlled and manipulated by control plane protocol operations which may result in route additions, removals and modifications. This also includes manipulations via the "static" and/or "direct" pseudo-protocols, see Section 5.3.1.

For every supported address family, exactly one RIB MUST be marked as the so-called default RIB. Its role is explained in Section 5.3.

Simple router implementations that do not advertise the feature "multiple-ribs" will typically create one system-controlled RIB per supported address family, and mark it as the default RIB.

More complex router implementations advertising the "multiple-ribs" feature support multiple RIBs per address family that can be used for policy routing and other purposes.

The following action (see Section 7.15 of [RFC7950]) is defined for the "rib" list:

- active-route -- return the active RIB route for the destination address that is specified as the action's input parameter.

5.3. Control Plane Protocol

The core routing data model provides an open-ended framework for defining multiple control plane protocol instances, e.g., for Layer 3 routing protocols. Each control plane protocol instance MUST be assigned a type, which is an identity derived from the "rt:control-plane-protocol" base identity. The core routing data model defines two identities for the direct and static pseudo-protocols (Section 5.3.1).

Multiple control plane protocol instances of the same type MAY be configured.

5.3.1. Routing Pseudo-Protocols

The core routing data model defines two special routing protocol types -- "direct" and "static". Both are in fact pseudo-protocols, which means that they are confined to the local device and do not exchange any routing information with adjacent routers.

Every implementation of the core routing data model MUST provide exactly one instance of the "direct" pseudo-protocol type. It is the source of direct routes for all configured address families. Direct routes are normally supplied by the operating system kernel, based on the configuration of network interface addresses, see Section 6.2.

A pseudo-protocol of the type "static" allows for specifying routes manually. It MAY be configured in zero or multiple instances, although a typical configuration will have exactly one instance.

5.3.2. Defining New Control Plane Protocols

It is expected that future YANG modules will create data models for additional control plane protocol types. Such a new module has to define the protocol-specific configuration and state data, and it has to integrate it into the core routing framework in the following way:

- A new identity MUST be defined for the control plane protocol and its base identity MUST be set to "rt:control-plane-protocol", or to an identity derived from "rt:control-plane-protocol".

- Additional route attributes MAY be defined, preferably in one place by means of defining a YANG grouping. The new attributes have to be inserted by augmenting the definitions of the nodes

  /rt:routing-state/rt:ribs/rt:rib/rt:routes/rt:route

  and

  /rt:routing-state/rt:ribs/rt:rib/rt:output/rt:route,

  and possibly other places in the configuration, state data, notifications, and input/output parameters of actions or RPC operations.

- Configuration parameters and/or state data for the new protocol can be defined by augmenting the "control-plane-protocol" data node under both "/routing" and "/routing-state".

By using a "when" statement, the augmented configuration parameters and state data specific to the new protocol SHOULD be made conditional and valid only if the value of "rt:type" or "rt:source-protocol" is equal to (or derived from) the new protocol's identity.

It is also RECOMMENDED that protocol-specific data nodes be encapsulated in an appropriately named container with presence. Such a container may contain mandatory data nodes that are otherwise forbidden at the top level of an augment.

The above steps are implemented by the example YANG module for the RIP routing protocol in Appendix C.

5.4. Parameters of IPv6 Router Advertisements

YANG module "ietf-ipv6-router-advertisements" (Section 9.1), which is a submodule of the "ietf-ipv6-unicast-routing" module, augments the configuration and state data of IPv6 interfaces with definitions of the following variables as required by [RFC4861], sec. 6.2.1:

- send-advertisements,
- max-rtr-adv-interval,
- min-rtr-adv-interval,
- managed-flag,
- other-config-flag,
- link-mtu,
- reachable-time,
- retrans-timer,
- cur-hop-limit,
- default-lifetime,
- prefix-list: a list of prefixes to be advertised.

  The following parameters are associated with each prefix in the list:

  * valid-lifetime,
  * on-link-flag,
  * preferred-lifetime,
  * autonomous-flag.

NOTES:

1. The "IsRouter" flag, which is also required by [RFC4861], is implemented in the "ietf-ip" module [RFC7277] (leaf "ip:forwarding").

2. The original specification [RFC4861] allows the implementations to decide whether the "valid-lifetime" and "preferred-lifetime" parameters remain the same in consecutive advertisements, or decrement in real time. However, the latter behavior seems problematic because the values might be reset again to the (higher) configured values after a configuration is reloaded. Moreover, no implementation is known to use the decrementing behavior. The "ietf-ipv6-router-advertisements" submodule therefore stipulates the former behavior with constant values.

6. Interactions with Other YANG Modules

The semantics of the core routing data model also depends on several configuration parameters that are defined in other YANG modules.

6.1. Module "ietf-interfaces"

The following boolean switch is defined in the "ietf-interfaces" YANG module [RFC7223]:

/if:interfaces/if:interface/if:enabled

If this switch is set to "false" for a network layer interface, then all routing and forwarding functions MUST be disabled on that interface.

6.2. Module "ietf-ip"

The following boolean switches are defined in the "ietf-ip" YANG module [RFC7277]:

/if:interfaces/if:interface/ip:ipv4/ip:enabled

If this switch is set to "false" for a network layer interface, then all IPv4 routing and forwarding functions MUST be disabled on that interface.

/if:interfaces/if:interface/ip:ipv4/ip:forwarding

If this switch is set to "false" for a network layer interface, then the forwarding of IPv4 datagrams through this interface MUST be disabled. However, the interface MAY participate in other IPv4 routing functions, such as routing protocols.

/if:interfaces/if:interface/ip:ipv6/ip:enabled

If this switch is set to "false" for a network layer interface, then all IPv6 routing and forwarding functions MUST be disabled on that interface.

/if:interfaces/if:interface/ip:ipv6/ip:forwarding

If this switch is set to "false" for a network layer interface, then the forwarding of IPv6 datagrams through this interface MUST be disabled. However, the interface MAY participate in other IPv6 routing functions, such as routing protocols.

In addition, the "ietf-ip" module allows for configuring IPv4 and IPv6 addresses and network prefixes or masks on network layer interfaces. Configuration of these parameters on an enabled interface MUST result in an immediate creation of the corresponding direct route. The destination prefix of this route is set according to the configured IP address and network prefix/mask, and the interface is set as the outgoing interface for that route.

7. Routing Management YANG Module

RFC Editor: In this section, replace all occurrences of 'XXXX' with the actual RFC number and all occurrences of the revision date below with the date of RFC publication (and remove this note).

<CODE BEGINS> file "ietf-routing@2016-11-03.yang"

```
module ietf-routing {

  yang-version "1.1";

  namespace "urn:ietf:params:xml:ns:yang:ietf-routing";

  prefix "rt";

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

  import ietf-interfaces {
    prefix "if";
  }

  organization
    "IETF NETMOD (NETCONF Data Modeling Language) Working Group";

  contact
    "WG Web:   <https://tools.ietf.org/wg/netmod/>
```

```
        WG List:  <mailto:netmod@ietf.org>

        WG Chair: Lou Berger
                  <mailto:lberger@labn.net>

        WG Chair: Kent Watsen
                  <mailto:kwatsen@juniper.net>

        Editor:   Ladislav Lhotka
                  <mailto:lhotka@nic.cz>

        Editor:   Acee Lindem
                  <mailto:acee@cisco.com>";

    description
      "This YANG module defines essential components for the management
       of a routing subsystem.

       Copyright (c) 2016 IETF Trust and the persons identified as
       authors of the code. All rights reserved.

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

       The key words 'MUST', 'MUST NOT', 'REQUIRED', 'SHALL', 'SHALL
       NOT', 'SHOULD', 'SHOULD NOT', 'RECOMMENDED', 'MAY', and
       'OPTIONAL' in the module text are to be interpreted as described
       in RFC 2119 (https://tools.ietf.org/html/rfc2119).

       This version of this YANG module is part of RFC XXXX
       (https://tools.ietf.org/html/rfcXXXX); see the RFC itself for
       full legal notices.";

    revision 2016-11-03 {
      description
        "Initial revision.";
      reference
        "RFC XXXX: A YANG Data Model for Routing Management";
    }

    /* Features */

    feature multiple-ribs {
      description
```

```
        "This feature indicates that the server supports user-defined
         RIBs.

         Servers that do not advertise this feature SHOULD provide
         exactly one system-controlled RIB per supported address family
         and make them also the default RIBs. These RIBs then appear as
         entries of the list /routing-state/ribs/rib.";
    }

    feature router-id {
      description
        "This feature indicates that the server supports configuration
         of an explicit 32-bit router ID that is used by some routing
         protocols.

         Servers that do not advertise this feature set a router ID
         algorithmically, usually to one of configured IPv4 addresses.
         However, this algorithm is implementation-specific.";
    }

    /* Identities */

    identity address-family {
      description
        "Base identity from which identities describing address
         families are derived.";
    }

    identity ipv4 {
      base address-family;
      description
        "This identity represents IPv4 address family.";
    }

    identity ipv6 {
      base address-family;
      description
        "This identity represents IPv6 address family.";
    }

    identity control-plane-protocol {
      description
        "Base identity from which control plane protocol identities are
         derived.";
    }

    identity routing-protocol {
      base control-plane-protocol;
```

```
    description
      "Identity from which Layer 3 routing protocol identities are
       derived.";
  }

  identity direct {
    base routing-protocol;
    description
      "Routing pseudo-protocol that provides routes to directly
       connected networks.";
  }

  identity static {
    base routing-protocol;
    description
      "Static routing pseudo-protocol.";
  }

  /* Type Definitions */

  typedef route-preference {
    type uint32;
    description
      "This type is used for route preferences.";
  }

  /* Groupings */

  grouping address-family {
    description
      "This grouping provides a leaf identifying an address
       family.";
    leaf address-family {
      type identityref {
        base address-family;
      }
      mandatory "true";
      description
        "Address family.";
    }
  }

  grouping router-id {
    description
      "This grouping provides router ID.";
    leaf router-id {
      type yang:dotted-quad;
      description
```

```
          "A 32-bit number in the form of a dotted quad that is used by
           some routing protocols identifying a router.";
        reference
          "RFC 2328: OSPF Version 2.";
      }
    }

    grouping special-next-hop {
      description
        "This grouping provides a leaf with an enumeration of special
         next-hops.";
      leaf special-next-hop {
        type enumeration {
          enum blackhole {
            description
              "Silently discard the packet.";
          }
          enum unreachable {
            description
              "Discard the packet and notify the sender with an error
               message indicating that the destination host is
               unreachable.";
          }
          enum prohibit {
            description
              "Discard the packet and notify the sender with an error
               message indicating that the communication is
               administratively prohibited.";
          }
          enum receive {
            description
              "The packet will be received by the local system.";
          }
        }
        description
          "Special next-hop options.";
      }
    }

    grouping next-hop-content {
      description
        "Generic parameters of next-hops in static routes.";
      choice next-hop-options {
        mandatory "true";
        description
          "Options for next-hops in static routes.

           It is expected that further cases will be added through
```

```
           augments from other modules.";
        case simple-next-hop {
          description
            "This case represents a simple next hop consisting of the
             next-hop address and/or outgoing interface.

             Modules for address families MUST augment this case with a
             leaf containing a next-hop address of that address
             family.";
          leaf outgoing-interface {
            type if:interface-ref;
            description
              "Name of the outgoing interface.";
          }
        }
        case special-next-hop {
          uses special-next-hop;
        }
        case next-hop-list {
          container next-hop-list {
            description
              "Container for multiple next-hops.";
            list next-hop {
              key "index";
              description
                "An entry of a next-hop list.

                 Modules for address families MUST augment this list
                 with a leaf containing a next-hop address of that
                 address family.";
              leaf index {
                type string;
                description
                  "An user-specified identifier utilised to uniquely
                   reference the next-hop entry in the next-hop list.
                   The value of this index has no semantic meaning
                   other than for referencing the entry.";
              }
              leaf outgoing-interface {
                type if:interface-ref;
                description
                  "Name of the outgoing interface.";
              }
            }
          }
        }
      }
    }
```

```
     grouping next-hop-state-content {
       description
         "Generic parameters of next-hops in state data.";
       choice next-hop-options {
         mandatory "true";
         description
           "Options for next-hops in state data.

            It is expected that further cases will be added through
            augments from other modules, e.g., for recursive
            next-hops.";
         case simple-next-hop {
           description
             "This case represents a simple next hop consisting of the
              next-hop address and/or outgoing interface.

              Modules for address families MUST augment this case with a
              leaf containing a next-hop address of that address
              family.";
           leaf outgoing-interface {
             type if:interface-state-ref;
             description
               "Name of the outgoing interface.";
           }
         }
         case special-next-hop {
           uses special-next-hop;
         }
         case next-hop-list {
           container next-hop-list {
             description
               "Container for multiple next-hops.";
             list next-hop {
               description
                 "An entry of a next-hop list.

                  Modules for address families MUST augment this list
                  with a leaf containing a next-hop address of that
                  address family.";
               leaf outgoing-interface {
                 type if:interface-state-ref;
                 description
                   "Name of the outgoing interface.";
               }
             }
           }
         }
       }
```

```
  }

  grouping route-metadata {
    description
      "Common route metadata.";
    leaf source-protocol {
      type identityref {
        base routing-protocol;
      }
      mandatory "true";
      description
        "Type of the routing protocol from which the route
         originated.";
    }
    leaf active {
      type empty;
      description
        "Presence of this leaf indicates that the route is preferred
         among all routes in the same RIB that have the same
         destination prefix.";
    }
    leaf last-updated {
      type yang:date-and-time;
      description
        "Time stamp of the last modification of the route. If the
         route was never modified, it is the time when the route was
         inserted into the RIB.";
    }
  }

  /* State data */

  container routing-state {
    config "false";
    description
      "State data of the routing subsystem.";
    uses router-id {
      description
        "Global router ID.

         It may be either configured or assigned algorithmically by
         the implementation.";
    }
    container interfaces {
      description
        "Network layer interfaces used for routing.";
      leaf-list interface {
        type if:interface-state-ref;
```

```
          description
            "Each entry is a reference to the name of a configured
             network layer interface.";
        }
      }
      container control-plane-protocols {
        description
          "Container for the list of routing protocol instances.";
        list control-plane-protocol {
          key "type name";
          description
            "State data of a control plane protocol instance.

             An implementation MUST provide exactly one
             system-controlled instance of the 'direct'
             pseudo-protocol. Instances of other control plane
             protocols MAY be created by configuration.";
          leaf type {
            type identityref {
              base control-plane-protocol;
            }
            description
              "Type of the control plane protocol.";
          }
          leaf name {
            type string;
            description
              "The name of the control plane protocol instance.

               For system-controlled instances this name is persistent,
               i.e., it SHOULD NOT change across reboots.";
          }
        }
      }
      container ribs {
        description
          "Container for RIBs.";
        list rib {
          key "name";
          min-elements "1";
          description
            "Each entry represents a RIB identified by the 'name' key.
             All routes in a RIB MUST belong to the same address
             family.

             An implementation SHOULD provide one system-controlled
             default RIB for each supported address family.";
          leaf name {
```

```
            type string;
            description
              "The name of the RIB.";
          }
          uses address-family;
          leaf default-rib {
            if-feature "multiple-ribs";
            type boolean;
            default "true";
            description
              "This flag has the value of 'true' if and only if the RIB
               is the default RIB for the given address family.

               By default, control plane protocols place their routes
               in the default RIBs.";
          }
          container routes {
            description
              "Current content of the RIB.";
            list route {
              description
                "A RIB route entry. This data node MUST be augmented
                 with information specific for routes of each address
                 family.";
              leaf route-preference {
                type route-preference;
                description
                  "This route attribute, also known as administrative
                   distance, allows for selecting the preferred route
                   among routes with the same destination prefix. A
                   smaller value means a more preferred route.";
              }
              container next-hop {
                description
                  "Route's next-hop attribute.";
                uses next-hop-state-content;
              }
              uses route-metadata;
            }
          }
          action active-route {
            description
              "Return the active RIB route that is used for the
               destination address.

               Address family specific modules MUST augment input
               parameters with a leaf named 'destination-address'.";
            output {
```

```
            container route {
              description
                "The active RIB route for the specified destination.

                 If no route exists in the RIB for the destination
                 address, no output is returned.

                 Address family specific modules MUST augment this
                 container with appropriate route contents.";
              container next-hop {
                description
                  "Route's next-hop attribute.";
                uses next-hop-state-content;
              }
              uses route-metadata;
            }
          }
        }
      }
    }

    /* Configuration Data */

    container routing {
      description
        "Configuration parameters for the routing subsystem.";
      uses router-id {
        if-feature "router-id";
        description
          "Configuration of the global router ID. Routing protocols
           that use router ID can use this parameter or override it
           with another value.";
      }
      container control-plane-protocols {
        description
          "Configuration of control plane protocol instances.";
        list control-plane-protocol {
          key "type name";
          description
            "Each entry contains configuration of a control plane
             protocol instance.";
          leaf type {
            type identityref {
              base control-plane-protocol;
            }
            description
              "Type of the control plane protocol - an identity derived
```

```
            from the 'control-plane-protocol' base identity.";
        }
        leaf name {
          type string;
          description
            "An arbitrary name of the control plane protocol
             instance.";
        }
        leaf description {
          type string;
          description
            "Textual description of the control plane protocol
             instance.";
        }
        container static-routes {
          when "derived-from-or-self(../type, 'rt:static')" {
            description
              "This container is only valid for the 'static' routing
               protocol.";
          }
          description
            "Configuration of the 'static' pseudo-protocol.

             Address-family-specific modules augment this node with
             their lists of routes.";
        }
      }
    }
    container ribs {
      description
        "Configuration of RIBs.";
      list rib {
        key "name";
        description
          "Each entry contains configuration for a RIB identified by
           the 'name' key.

           Entries having the same key as a system-controlled entry
           of the list /routing-state/ribs/rib are used for
           configuring parameters of that entry. Other entries define
           additional user-controlled RIBs.";
        leaf name {
          type string;
          description
            "The name of the RIB.

             For system-controlled entries, the value of this leaf
             must be the same as the name of the corresponding entry
```

```
                in state data.

                For user-controlled entries, an arbitrary name can be
                used.";
            }
            uses address-family {
              description
                "Address family of the RIB.

                 It is mandatory for user-controlled RIBs. For
                 system-controlled RIBs it can be omitted, otherwise it
                 must match the address family of the corresponding state
                 entry.";
              refine "address-family" {
                mandatory "false";
              }
            }
            leaf description {
              type string;
              description
                "Textual description of the RIB.";
            }
          }
        }
      }
    }
```

<CODE ENDS>

8. IPv4 Unicast Routing Management YANG Module

RFC Editor: In this section, replace all occurrences of 'XXXX' with the actual RFC number and all occurrences of the revision date below with the date of RFC publication (and remove this note).

<CODE BEGINS> file "ietf-ipv4-unicast-routing@2016-11-03.yang"

```
module ietf-ipv4-unicast-routing {

  yang-version "1.1";

  namespace "urn:ietf:params:xml:ns:yang:ietf-ipv4-unicast-routing";

  prefix "v4ur";

  import ietf-routing {
    prefix "rt";
  }
```

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

  organization
    "IETF NETMOD (NETCONF Data Modeling Language) Working Group";

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

     WG Chair: Lou Berger
               <mailto:lberger@labn.net>

     WG Chair: Kent Watsen
               <mailto:kwatsen@juniper.net>

     Editor:   Ladislav Lhotka
               <mailto:lhotka@nic.cz>

     Editor:   Acee Lindem
               <mailto:acee@cisco.com>";

  description
    "This YANG module augments the 'ietf-routing' module with basic
     configuration and state data for IPv4 unicast routing.

     Copyright (c) 2016 IETF Trust and the persons identified as
     authors of the code. All rights reserved.

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

     The key words 'MUST', 'MUST NOT', 'REQUIRED', 'SHALL', 'SHALL
     NOT', 'SHOULD', 'SHOULD NOT', 'RECOMMENDED', 'MAY', and
     'OPTIONAL' in the module text are to be interpreted as described
     in RFC 2119 (https://tools.ietf.org/html/rfc2119).

     This version of this YANG module is part of RFC XXXX
     (https://tools.ietf.org/html/rfcXXXX); see the RFC itself for
     full legal notices.";

  revision 2016-11-03 {
    description
```

```
        "Initial revision.";
      reference
        "RFC XXXX: A YANG Data Model for Routing Management";
    }

    /* Identities */

    identity ipv4-unicast {
      base rt:ipv4;
      description
        "This identity represents the IPv4 unicast address family.";
    }

    /* State data */

    augment "/rt:routing-state/rt:ribs/rt:rib/rt:routes/rt:route" {
      when "derived-from-or-self(../../rt:address-family, "
         + "'v4ur:ipv4-unicast')" {
        description
          "This augment is valid only for IPv4 unicast.";
      }
      description
        "This leaf augments an IPv4 unicast route.";
      leaf destination-prefix {
        type inet:ipv4-prefix;
        description
          "IPv4 destination prefix.";
      }
    }

    augment "/rt:routing-state/rt:ribs/rt:rib/rt:routes/rt:route/"
          + "rt:next-hop/rt:next-hop-options/rt:simple-next-hop" {
      when "derived-from-or-self(../../../rt:address-family, "
         + "'v4ur:ipv4-unicast')" {
        description
          "This augment is valid only for IPv4 unicast.";
      }
      description
        "Augment 'simple-next-hop' case in IPv4 unicast routes.";
      leaf next-hop-address {
        type inet:ipv4-address;
        description
          "IPv4 address of the next-hop.";
      }
    }

    augment "/rt:routing-state/rt:ribs/rt:rib/rt:routes/rt:route/"
          + "rt:next-hop/rt:next-hop-options/rt:next-hop-list/"
```

```
          + "rt:next-hop-list/rt:next-hop" {
        when "derived-from-or-self(../../../../../rt:address-family, "
           + "'v4ur:ipv4-unicast')" {
          description
            "This augment is valid only for IPv4 unicast.";
        }
        description
          "This leaf augments the 'next-hop-list' case of IPv4 unicast
           routes.";
        leaf address {
          type inet:ipv4-address;
          description
            "IPv4 address of the next-hop.";
        }
      }

      augment
        "/rt:routing-state/rt:ribs/rt:rib/rt:active-route/rt:input" {
        when "derived-from-or-self(../rt:address-family, "
           + "'v4ur:ipv4-unicast')" {
          description
            "This augment is valid only for IPv4 unicast RIBs.";
        }
        description
          "This augment adds the input parameter of the 'active-route'
           action.";
        leaf destination-address {
          type inet:ipv4-address;
          description
            "IPv4 destination address.";
        }
      }

      augment "/rt:routing-state/rt:ribs/rt:rib/rt:active-route/"
            + "rt:output/rt:route" {
        when "derived-from-or-self(../../rt:address-family, "
           + "'v4ur:ipv4-unicast')" {
          description
            "This augment is valid only for IPv4 unicast.";
        }
        description
          "This augment adds the destination prefix to the reply of the
           'active-route' action.";
        leaf destination-prefix {
          type inet:ipv4-prefix;
          description
            "IPv4 destination prefix.";
        }
```

```
  }

  augment "/rt:routing-state/rt:ribs/rt:rib/rt:active-route/"
        + "rt:output/rt:route/rt:next-hop/rt:next-hop-options/"
        + "rt:simple-next-hop" {
    when "derived-from-or-self(../../../rt:address-family, "
       + "'v4ur:ipv4-unicast')" {
      description
        "This augment is valid only for IPv4 unicast.";
    }
    description
      "Augment 'simple-next-hop' case in the reply to the
       'active-route' action.";
    leaf next-hop-address {
      type inet:ipv4-address;
      description
        "IPv4 address of the next-hop.";
    }
  }

  augment "/rt:routing-state/rt:ribs/rt:rib/rt:active-route/"
        + "rt:output/rt:route/rt:next-hop/rt:next-hop-options/"
        + "rt:next-hop-list/rt:next-hop-list/rt:next-hop" {
    when "derived-from-or-self(../../../../../rt:address-family, "
       + "'v4ur:ipv4-unicast')" {
      description
        "This augment is valid only for IPv4 unicast.";
    }
    description
      "Augment 'next-hop-list' case in the reply to the
       'active-route' action.";
    leaf next-hop-address {
      type inet:ipv4-address;
      description
        "IPv4 address of the next-hop.";
    }
  }

  /* Configuration data */

  augment "/rt:routing/rt:control-plane-protocols/"
        + "rt:control-plane-protocol/rt:static-routes" {
    description
      "This augment defines the configuration of the 'static'
       pseudo-protocol with data specific to IPv4 unicast.";
    container ipv4 {
      description
        "Configuration of a 'static' pseudo-protocol instance
```

```
           consists of a list of routes.";
         list route {
           key "destination-prefix";
           description
             "A list of static routes.";
           leaf destination-prefix {
             type inet:ipv4-prefix;
             mandatory "true";
             description
               "IPv4 destination prefix.";
           }
           leaf description {
             type string;
             description
               "Textual description of the route.";
           }
           container next-hop {
             description
               "Configuration of next-hop.";
             uses rt:next-hop-content {
               augment "next-hop-options/simple-next-hop" {
                 description
                   "Augment 'simple-next-hop' case in IPv4 static
                    routes.";
                 leaf next-hop-address {
                   type inet:ipv4-address;
                   description
                     "IPv4 address of the next-hop.";
                 }
               }
               augment "next-hop-options/next-hop-list/next-hop-list/"
                     + "next-hop" {
                 description
                   "Augment 'next-hop-list' case in IPv4 static
                    routes.";
                 leaf next-hop-address {
                   type inet:ipv4-address;
                   description
                     "IPv4 address of the next-hop.";
                 }
               }
             }
           }
         }
       }
     }
   }
```

<CODE ENDS>

9. IPv6 Unicast Routing Management YANG Module

RFC Editor: In this section, replace all occurrences of 'XXXX' with the actual RFC number and all occurrences of the revision date below with the date of RFC publication (and remove this note).

<CODE BEGINS> file "ietf-ipv6-unicast-routing@2016-11-03.yang"

```
module ietf-ipv6-unicast-routing {

  yang-version "1.1";

  namespace "urn:ietf:params:xml:ns:yang:ietf-ipv6-unicast-routing";

  prefix "v6ur";

  import ietf-routing {
    prefix "rt";
  }

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

  include ietf-ipv6-router-advertisements {
    revision-date 2016-11-03;
  }

  organization
    "IETF NETMOD (NETCONF Data Modeling Language) Working Group";

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

     WG Chair: Lou Berger
               <mailto:lberger@labn.net>

     WG Chair: Kent Watsen
               <mailto:kwatsen@juniper.net>

     Editor:   Ladislav Lhotka
               <mailto:lhotka@nic.cz>

     Editor:   Acee Lindem
               <mailto:acee@cisco.com>";
```

```
  description
    "This YANG module augments the 'ietf-routing' module with basic
     configuration and state data for IPv6 unicast routing.

     Copyright (c) 2016 IETF Trust and the persons identified as
     authors of the code. All rights reserved.

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

     The key words 'MUST', 'MUST NOT', 'REQUIRED', 'SHALL', 'SHALL
     NOT', 'SHOULD', 'SHOULD NOT', 'RECOMMENDED', 'MAY', and
     'OPTIONAL' in the module text are to be interpreted as described
     in RFC 2119 (https://tools.ietf.org/html/rfc2119).

     This version of this YANG module is part of RFC XXXX
     (https://tools.ietf.org/html/rfcXXXX); see the RFC itself for
     full legal notices.";

  revision 2016-11-03 {
    description
      "Initial revision.";
    reference
      "RFC XXXX: A YANG Data Model for Routing Management";
  }

  /* Identities */

  identity ipv6-unicast {
    base rt:ipv6;
    description
      "This identity represents the IPv6 unicast address family.";
  }

  /* State data */

  augment "/rt:routing-state/rt:ribs/rt:rib/rt:routes/rt:route" {
    when "derived-from-or-self(../../rt:address-family, "
       + "'v6ur:ipv6-unicast')" {
      description
        "This augment is valid only for IPv6 unicast.";
    }
    description
      "This leaf augments an IPv6 unicast route.";
```

```
      leaf destination-prefix {
        type inet:ipv6-prefix;
        description
          "IPv6 destination prefix.";
      }
    }

    augment "/rt:routing-state/rt:ribs/rt:rib/rt:routes/rt:route/"
          + "rt:next-hop/rt:next-hop-options/rt:simple-next-hop" {
      when "derived-from-or-self(../../../rt:address-family, "
         + "'v6ur:ipv6-unicast')" {
        description
          "This augment is valid only for IPv6 unicast.";
      }
      description
        "Augment 'simple-next-hop' case in IPv6 unicast routes.";
      leaf next-hop-address {
        type inet:ipv6-address;
        description
          "IPv6 address of the next-hop.";
      }
    }

    augment "/rt:routing-state/rt:ribs/rt:rib/rt:routes/rt:route/"
          + "rt:next-hop/rt:next-hop-options/rt:next-hop-list/"
          + "rt:next-hop-list/rt:next-hop" {
      when "derived-from-or-self(../../../../../rt:address-family, "
         + "'v6ur:ipv6-unicast')" {
        description
          "This augment is valid only for IPv6 unicast.";
      }
      description
        "This leaf augments the 'next-hop-list' case of IPv6 unicast
         routes.";
      leaf address {
        type inet:ipv6-address;
        description
          "IPv6 address of the next-hop.";
      }
    }

    augment
      "/rt:routing-state/rt:ribs/rt:rib/rt:active-route/rt:input" {
      when "derived-from-or-self(../rt:address-family, "
         + "'v6ur:ipv6-unicast')" {
        description
          "This augment is valid only for IPv6 unicast RIBs.";
      }
```

```
      description
        "This augment adds the input parameter of the 'active-route'
         action.";
      leaf destination-address {
        type inet:ipv6-address;
        description
          "IPv6 destination address.";
      }
    }

    augment "/rt:routing-state/rt:ribs/rt:rib/rt:active-route/"
          + "rt:output/rt:route" {
      when "derived-from-or-self(../../rt:address-family, "
         + "'v6ur:ipv6-unicast')" {
        description
          "This augment is valid only for IPv6 unicast.";
      }
      description
        "This augment adds the destination prefix to the reply of the
         'active-route' action.";
      leaf destination-prefix {
        type inet:ipv6-prefix;
        description
          "IPv6 destination prefix.";
      }
    }

    augment "/rt:routing-state/rt:ribs/rt:rib/rt:active-route/"
          + "rt:output/rt:route/rt:next-hop/rt:next-hop-options/"
          + "rt:simple-next-hop" {
      when "derived-from-or-self(../../../rt:address-family, "
         + "'v6ur:ipv6-unicast')" {
        description
          "This augment is valid only for IPv6 unicast.";
      }
      description
        "Augment 'simple-next-hop' case in the reply to the
         'active-route' action.";
      leaf next-hop-address {
        type inet:ipv6-address;
        description
          "IPv6 address of the next-hop.";
      }
    }

    augment "/rt:routing-state/rt:ribs/rt:rib/rt:active-route/"
          + "rt:output/rt:route/rt:next-hop/rt:next-hop-options/"
          + "rt:next-hop-list/rt:next-hop-list/rt:next-hop" {
```

```
      when "derived-from-or-self(../../../../../rt:address-family, "
         + "'v6ur:ipv6-unicast')" {
        description
          "This augment is valid only for IPv6 unicast.";
      }
      description
        "Augment 'next-hop-list' case in the reply to the
         'active-route' action.";
      leaf next-hop-address {
        type inet:ipv6-address;
        description
          "IPv6 address of the next-hop.";
      }
    }

    /* Configuration data */

    augment "/rt:routing/rt:control-plane-protocols/"
          + "rt:control-plane-protocol/rt:static-routes" {
      description
        "This augment defines the configuration of the 'static'
         pseudo-protocol with data specific to IPv6 unicast.";
      container ipv6 {
        description
          "Configuration of a 'static' pseudo-protocol instance
           consists of a list of routes.";
        list route {
          key "destination-prefix";
          description
            "A list of static routes.";
          leaf destination-prefix {
            type inet:ipv6-prefix;
            mandatory "true";
            description
              "IPv6 destination prefix.";
          }
          leaf description {
            type string;
            description
              "Textual description of the route.";
          }
          container next-hop {
            description
              "Configuration of next-hop.";
            uses rt:next-hop-content {
              augment "next-hop-options/simple-next-hop" {
                description
                  "Augment 'simple-next-hop' case in IPv6 static
```

```
                   routes.";
                leaf next-hop-address {
                  type inet:ipv6-address;
                  description
                    "IPv6 address of the next-hop.";
                }
              }
              augment "next-hop-options/next-hop-list/next-hop-list/"
                    + "next-hop" {
                description
                  "Augment 'next-hop-list' case in IPv6 static
                   routes.";
                leaf next-hop-address {
                  type inet:ipv6-address;
                  description
                    "IPv6 address of the next-hop.";
                }
              }
            }
          }
        }
      }
    }
  }
```

<CODE ENDS>

9.1. IPv6 Router Advertisements Submodule

RFC Editor: In this section, replace all occurrences of 'XXXX' with the actual RFC number and all occurrences of the revision date below with the date of RFC publication (and remove this note).

<CODE BEGINS> file "ietf-ipv6-router-advertisements@2016-11-03.yang"

```
submodule ietf-ipv6-router-advertisements {

  yang-version "1.1";

  belongs-to ietf-ipv6-unicast-routing {
    prefix "v6ur";
  }

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

  import ietf-interfaces {
```

```
  prefix "if";
}

import ietf-ip {
  prefix "ip";
}

organization
  "IETF NETMOD (NETCONF Data Modeling Language) Working Group";

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

   WG Chair: Lou Berger
             <mailto:lberger@labn.net>

   WG Chair: Kent Watsen
             <mailto:kwatsen@juniper.net>

   Editor:   Ladislav Lhotka
             <mailto:lhotka@nic.cz>

   Editor:   Acee Lindem
             <mailto:acee@cisco.com>";

description
  "This YANG module augments the 'ietf-ip' module with
   configuration and state data of IPv6 router advertisements.

   Copyright (c) 2016 IETF Trust and the persons identified as
   authors of the code. All rights reserved.

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

   The key words 'MUST', 'MUST NOT', 'REQUIRED', 'SHALL', 'SHALL
   NOT', 'SHOULD', 'SHOULD NOT', 'RECOMMENDED', 'MAY', and
   'OPTIONAL' in the module text are to be interpreted as described
   in RFC 2119 (https://tools.ietf.org/html/rfc2119).

   This version of this YANG module is part of RFC XXXX
   (https://tools.ietf.org/html/rfcXXXX); see the RFC itself for
   full legal notices.";
```

```
     reference
       "RFC 4861: Neighbor Discovery for IP version 6 (IPv6).";

     revision 2016-11-03 {
       description
         "Initial revision.";
       reference
         "RFC XXXX: A YANG Data Model for Routing Management";
     }

     /* State data */

     augment "/if:interfaces-state/if:interface/ip:ipv6" {
       description
         "Augment interface state data with parameters of IPv6 router
          advertisements.";
       container ipv6-router-advertisements {
         description
           "Parameters of IPv6 Router Advertisements.";
         leaf send-advertisements {
           type boolean;
           description
             "A flag indicating whether or not the router sends periodic
              Router Advertisements and responds to Router
              Solicitations.";
         }
         leaf max-rtr-adv-interval {
           type uint16 {
             range "4..1800";
           }
           units "seconds";
           description
             "The maximum time allowed between sending unsolicited
              multicast Router Advertisements from the interface.";
         }
         leaf min-rtr-adv-interval {
           type uint16 {
             range "3..1350";
           }
           units "seconds";
           description
             "The minimum time allowed between sending unsolicited
              multicast Router Advertisements from the interface.";
         }
         leaf managed-flag {
           type boolean;
           description
             "The value that is placed in the 'Managed address
```

```
          configuration' flag field in the Router Advertisement.";
        }
        leaf other-config-flag {
          type boolean;
          description
            "The value that is placed in the 'Other configuration' flag
             field in the Router Advertisement.";
        }
        leaf link-mtu {
          type uint32;
          description
            "The value that is placed in MTU options sent by the
             router. A value of zero indicates that no MTU options are
             sent.";
        }
        leaf reachable-time {
          type uint32 {
            range "0..3600000";
          }
          units "milliseconds";
          description
            "The value that is placed in the Reachable Time field in
             the Router Advertisement messages sent by the router. A
             value of zero means unspecified (by this router).";
        }
        leaf retrans-timer {
          type uint32;
          units "milliseconds";
          description
            "The value that is placed in the Retrans Timer field in the
             Router Advertisement messages sent by the router. A value
             of zero means unspecified (by this router).";
        }
        leaf cur-hop-limit {
          type uint8;
          description
            "The value that is placed in the Cur Hop Limit field in the
             Router Advertisement messages sent by the router. A value
             of zero means unspecified (by this router).";
        }
        leaf default-lifetime {
          type uint16 {
            range "0..9000";
          }
          units "seconds";
          description
            "The value that is placed in the Router Lifetime field of
             Router Advertisements sent from the interface, in seconds.
```

```
            A value of zero indicates that the router is not to be
            used as a default router.";
        }
        container prefix-list {
          description
            "A list of prefixes that are placed in Prefix Information
             options in Router Advertisement messages sent from the
             interface.

             By default, these are all prefixes that the router
             advertises via routing protocols as being on-link for the
             interface from which the advertisement is sent.";
          list prefix {
            key "prefix-spec";
            description
              "Advertised prefix entry and its parameters.";
            leaf prefix-spec {
              type inet:ipv6-prefix;
              description
                "IPv6 address prefix.";
            }
            leaf valid-lifetime {
              type uint32;
              units "seconds";
              description
                "The value that is placed in the Valid Lifetime in the
                 Prefix Information option. The designated value of all
                 1's (0xffffffff) represents infinity.

                 An implementation SHOULD keep this value constant in
                 consecutive advertisements except when it is
                 explicitly changed in configuration.";
            }
            leaf on-link-flag {
              type boolean;
              description
                "The value that is placed in the on-link flag ('L-bit')
                 field in the Prefix Information option.";
            }
            leaf preferred-lifetime {
              type uint32;
              units "seconds";
              description
                "The value that is placed in the Preferred Lifetime in
                 the Prefix Information option, in seconds. The
                 designated value of all 1's (0xffffffff) represents
                 infinity.
```

```
                An implementation SHOULD keep this value constant in
                consecutive advertisements except when it is
                explicitly changed in configuration.";
            }
            leaf autonomous-flag {
              type boolean;
              description
                "The value that is placed in the Autonomous Flag field
                 in the Prefix Information option.";
            }
          }
        }
      }
    }

    /* Configuration data */

    augment "/if:interfaces/if:interface/ip:ipv6" {
      description
        "Augment interface configuration with parameters of IPv6 router
         advertisements.";
      container ipv6-router-advertisements {
        description
          "Configuration of IPv6 Router Advertisements.";
        leaf send-advertisements {
          type boolean;
          default "false";
          description
            "A flag indicating whether or not the router sends periodic
             Router Advertisements and responds to Router
             Solicitations.";
          reference
            "RFC 4861: Neighbor Discovery for IP version 6 (IPv6) -
             AdvSendAdvertisements.";
        }
        leaf max-rtr-adv-interval {
          type uint16 {
            range "4..1800";
          }
          units "seconds";
          default "600";
          description
            "The maximum time allowed between sending unsolicited
             multicast Router Advertisements from the interface.";
          reference
            "RFC 4861: Neighbor Discovery for IP version 6 (IPv6) -
             MaxRtrAdvInterval.";
        }
```

```
        leaf min-rtr-adv-interval {
          type uint16 {
            range "3..1350";
          }
          units "seconds";
          must ". <= 0.75 * ../max-rtr-adv-interval" {
            description
              "The value MUST NOT be greater than 75 % of
               'max-rtr-adv-interval'.";
          }
          description
            "The minimum time allowed between sending unsolicited
             multicast Router Advertisements from the interface.

             The default value to be used operationally if this leaf is
             not configured is determined as follows:

             - if max-rtr-adv-interval >= 9 seconds, the default value
               is 0.33 * max-rtr-adv-interval;

             - otherwise it is 0.75 * max-rtr-adv-interval.";
          reference
            "RFC 4861: Neighbor Discovery for IP version 6 (IPv6) -
             MinRtrAdvInterval.";
        }
        leaf managed-flag {
          type boolean;
          default "false";
          description
            "The value to be placed in the 'Managed address
             configuration' flag field in the Router Advertisement.";
          reference
            "RFC 4861: Neighbor Discovery for IP version 6 (IPv6) -
             AdvManagedFlag.";
        }
        leaf other-config-flag {
          type boolean;
          default "false";
          description
            "The value to be placed in the 'Other configuration' flag
             field in the Router Advertisement.";
          reference
            "RFC 4861: Neighbor Discovery for IP version 6 (IPv6) -
             AdvOtherConfigFlag.";
        }
        leaf link-mtu {
          type uint32;
          default "0";
```

```
          description
            "The value to be placed in MTU options sent by the router.
             A value of zero indicates that no MTU options are sent.";
          reference
            "RFC 4861: Neighbor Discovery for IP version 6 (IPv6) -
             AdvLinkMTU.";
        }
        leaf reachable-time {
          type uint32 {
            range "0..3600000";
          }
          units "milliseconds";
          default "0";
          description
            "The value to be placed in the Reachable Time field in the
             Router Advertisement messages sent by the router. A value
             of zero means unspecified (by this router).";
          reference
            "RFC 4861: Neighbor Discovery for IP version 6 (IPv6) -
             AdvReachableTime.";
        }
        leaf retrans-timer {
          type uint32;
          units "milliseconds";
          default "0";
          description
            "The value to be placed in the Retrans Timer field in the
             Router Advertisement messages sent by the router. A value
             of zero means unspecified (by this router).";
          reference
            "RFC 4861: Neighbor Discovery for IP version 6 (IPv6) -
             AdvRetransTimer.";
        }
        leaf cur-hop-limit {
          type uint8;
          description
            "The value to be placed in the Cur Hop Limit field in the
             Router Advertisement messages sent by the router. A value
             of zero means unspecified (by this router).

             If this parameter is not configured, the device SHOULD use
             the value specified in IANA Assigned Numbers that was in
             effect at the time of implementation.";
          reference
            "RFC 4861: Neighbor Discovery for IP version 6 (IPv6) -
             AdvCurHopLimit.

             IANA: IP Parameters,
```

```
          http://www.iana.org/assignments/ip-parameters";
        }
        leaf default-lifetime {
          type uint16 {
            range "0..9000";
          }
          units "seconds";
          description
            "The value to be placed in the Router Lifetime field of
             Router Advertisements sent from the interface, in seconds.
             It MUST be either zero or between max-rtr-adv-interval and
             9000 seconds. A value of zero indicates that the router is
             not to be used as a default router. These limits may be
             overridden by specific documents that describe how IPv6
             operates over different link layers.

             If this parameter is not configured, the device SHOULD use
             a value of 3 * max-rtr-adv-interval.";
          reference
            "RFC 4861: Neighbor Discovery for IP version 6 (IPv6) -
             AdvDefaultLifeTime.";
        }
        container prefix-list {
          description
            "Configuration of prefixes to be placed in Prefix
             Information options in Router Advertisement messages sent
             from the interface.

             Prefixes that are advertised by default but do not have
             their entries in the child 'prefix' list are advertised
             with the default values of all parameters.

             The link-local prefix SHOULD NOT be included in the list
             of advertised prefixes.";
          reference
            "RFC 4861: Neighbor Discovery for IP version 6 (IPv6) -
             AdvPrefixList.";
          list prefix {
            key "prefix-spec";
            description
              "Configuration of an advertised prefix entry.";
            leaf prefix-spec {
              type inet:ipv6-prefix;
              description
                "IPv6 address prefix.";
            }
            choice control-adv-prefixes {
              default "advertise";
```

```
            description
              "The prefix either may be explicitly removed from the
               set of advertised prefixes, or parameters with which
               it is advertised may be specified (default case).";
            leaf no-advertise {
              type empty;
              description
                "The prefix will not be advertised.

                 This can be used for removing the prefix from the
                 default set of advertised prefixes.";
            }
            case advertise {
              leaf valid-lifetime {
                type uint32;
                units "seconds";
                default "2592000";
                description
                  "The value to be placed in the Valid Lifetime in
                   the Prefix Information option. The designated
                   value of all 1's (0xffffffff) represents
                   infinity.";
                reference
                  "RFC 4861: Neighbor Discovery for IP version 6
                   (IPv6) - AdvValidLifetime.";
              }
              leaf on-link-flag {
                type boolean;
                default "true";
                description
                  "The value to be placed in the on-link flag
                   ('L-bit') field in the Prefix Information
                   option.";
                reference
                  "RFC 4861: Neighbor Discovery for IP version 6
                   (IPv6) - AdvOnLinkFlag.";
              }
              leaf preferred-lifetime {
                type uint32;
                units "seconds";
                must ". <= ../valid-lifetime" {
                  description
                    "This value MUST NOT be greater than
                     valid-lifetime.";
                }
                default "604800";
                description
                  "The value to be placed in the Preferred Lifetime
```

```
                    in the Prefix Information option. The designated
                    value of all 1's (0xffffffff) represents
                    infinity.";
                  reference
                    "RFC 4861: Neighbor Discovery for IP version 6
                     (IPv6) - AdvPreferredLifetime.";
                }
                leaf autonomous-flag {
                  type boolean;
                  default "true";
                  description
                    "The value to be placed in the Autonomous Flag
                     field in the Prefix Information option.";
                  reference
                    "RFC 4861: Neighbor Discovery for IP version 6
                     (IPv6) - AdvAutonomousFlag.";
                }
              }
            }
          }
        }
      }
    }
  }
```

<CODE ENDS>

10. IANA Considerations

RFC Ed.: In this section, replace all occurrences of 'XXXX' with the actual RFC number (and remove this note).

This document registers the following namespace URIs in the IETF XML registry [RFC3688]:

URI: urn:ietf:params:xml:ns:yang:ietf-routing

Registrant Contact: The IESG.

XML: N/A, the requested URI is an XML namespace.

```
---------------------------------------------------------------------
URI: urn:ietf:params:xml:ns:yang:ietf-ipv4-unicast-routing

Registrant Contact: The IESG.

XML: N/A, the requested URI is an XML namespace.
---------------------------------------------------------------------

---------------------------------------------------------------------
URI: urn:ietf:params:xml:ns:yang:ietf-ipv6-unicast-routing

Registrant Contact: The IESG.

XML: N/A, the requested URI is an XML namespace.
---------------------------------------------------------------------
```

This document registers the following YANG modules in the YANG Module Names registry [RFC6020]:

```
---------------------------------------------------------------------
name:         ietf-routing
namespace:    urn:ietf:params:xml:ns:yang:ietf-routing
prefix:       rt
reference:    RFC XXXX
---------------------------------------------------------------------

---------------------------------------------------------------------
name:         ietf-ipv4-unicast-routing
namespace:    urn:ietf:params:xml:ns:yang:ietf-ipv4-unicast-routing
prefix:       v4ur
reference:    RFC XXXX
---------------------------------------------------------------------

---------------------------------------------------------------------
name:         ietf-ipv6-unicast-routing
namespace:    urn:ietf:params:xml:ns:yang:ietf-ipv6-unicast-routing
prefix:       v6ur
reference:    RFC XXXX
---------------------------------------------------------------------
```

This document registers the following YANG submodule in the YANG Module Names registry [RFC6020]:

```
---------------------------------------------------------------------
name:         ietf-ipv6-router-advertisements
parent:       ietf-ipv6-unicast-routing
reference:    RFC XXXX
---------------------------------------------------------------------
```

11. Security Considerations

Configuration and state data conforming to the core routing data model (defined in this document) are designed to be accessed via a management protocol with secure transport layer, such as NETCONF [RFC6241]. The NETCONF access control model [RFC6536] provides the means to restrict access for particular NETCONF users to a pre-configured subset of all available NETCONF protocol operations and content.

A number of configuration data nodes defined in the YANG modules belonging to the core routing data model are writable/creatable/deletable (i.e., "config true" in YANG terms, which is the default). These data nodes may be considered sensitive or vulnerable in some network environments. Write operations to these data nodes, such as "edit-config" in NETCONF, can have negative effects on the network if the protocol operations are not properly protected.

The vulnerable "config true" parameters and subtrees are the following:

/routing/control-plane-protocols/control-plane-protocol: This list specifies the control plane protocols configured on a device.

/routing/ribs/rib: This list specifies the RIBs configured for the device.

Unauthorised access to any of these lists can adversely affect the routing subsystem of both the local device and the network. This may lead to network malfunctions, delivery of packets to inappropriate destinations and other problems.

12. Acknowledgments

The authors wish to thank Nitin Bahadur, Martin Bjorklund, Dean Bogdanovic, Jeff Haas, Joel Halpern, Wes Hardaker, Sriganesh Kini, David Lamparter, Andrew McGregor, Jan Medved, Xiang Li, Stephane Litkowski, Thomas Morin, Tom Petch, Yingzhen Qu, Bruno Rijsman, Juergen Schoenwaelder, Phil Shafer, Dave Thaler, Yi Yang, Derek Man-Kit Yeung and Jeffrey Zhang for their helpful comments and suggestions.

13. References

13.1. Normative References

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

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

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

[RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <http://www.rfc-editor.org/info/rfc6020>.

[RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <http://www.rfc-editor.org/info/rfc6241>.

[RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, DOI 10.17487/RFC6991, July 2013, <http://www.rfc-editor.org/info/rfc6991>.

[RFC7223] Bjorklund, M., "A YANG Data Model for Interface Management", RFC 7223, DOI 10.17487/RFC7223, May 2014, <http://www.rfc-editor.org/info/rfc7223>.

[RFC7277] Bjorklund, M., "A YANG Data Model for IP Management", RFC 7277, DOI 10.17487/RFC7277, June 2014, <http://www.rfc-editor.org/info/rfc7277>.

[RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <http://www.rfc-editor.org/info/rfc7950>.

13.2. Informative References

[RFC6087] Bierman, A., "Guidelines for Authors and Reviewers of YANG Data Model Documents", RFC 6087, DOI 10.17487/RFC6087, January 2011, <http://www.rfc-editor.org/info/rfc6087>.

[RFC6536] Bierman, A. and M. Bjorklund, "Network Configuration Protocol (NETCONF) Access Control Model", RFC 6536, DOI 10.17487/RFC6536, March 2012, <http://www.rfc-editor.org/info/rfc6536>.

[RFC7895] Bierman, A., Bjorklund, M., and K. Watsen, "YANG Module Library", RFC 7895, DOI 10.17487/RFC7895, June 2016, <http://www.rfc-editor.org/info/rfc7895>.

[RFC7951] Lhotka, L., "JSON Encoding of Data Modeled with YANG", RFC 7951, DOI 10.17487/RFC7951, August 2016, <http://www.rfc-editor.org/info/rfc7951>.

Appendix A. The Complete Data Trees

This appendix presents the complete configuration and state data trees of the core routing data model. See Section 2.2 for an explanation of the symbols used. Data type of every leaf node is shown near the right end of the corresponding line.

A.1. Configuration Data

```
   +--rw routing
      +--rw router-id?                 yang:dotted-quad
      +--rw control-plane-protocols
      |  +--rw control-plane-protocol* [type name]
      |     +--rw type             identityref
      |     +--rw name             string
      |     +--rw description?     string
      |     +--rw static-routes
      |        +--rw v6ur:ipv6
      |        |  +--rw v6ur:route* [destination-prefix]
      |        |     +--rw v6ur:destination-prefix    inet:ipv6-prefix
      |        |     +--rw v6ur:description?          string
      |        |     +--rw v6ur:next-hop
      |        |        +--rw (v6ur:next-hop-options)
      |        |           +--:(v6ur:simple-next-hop)
      |        |           |  +--rw v6ur:outgoing-interface?
      |        |           |  +--rw v6ur:next-hop-address?
      |        |           +--:(v6ur:special-next-hop)
      |        |           |  +--rw v6ur:special-next-hop?   enumeration
      |        |           +--:(v6ur:next-hop-list)
      |        |              +--rw v6ur:next-hop-list
      |        |                 +--rw v6ur:next-hop* [index]
      |        |                    +--rw v6ur:index              string
      |        |                    +--rw v6ur:outgoing-interface?
      |        |                    +--rw v6ur:next-hop-address?
      |        +--rw v4ur:ipv4
      |           +--rw v4ur:route* [destination-prefix]
      |              +--rw v4ur:destination-prefix    inet:ipv4-prefix
      |              +--rw v4ur:description?          string
      |              +--rw v4ur:next-hop
      |                 +--rw (v4ur:next-hop-options)
      |                    +--:(v4ur:simple-next-hop)
      |                    |  +--rw v4ur:outgoing-interface?
      |                    |  +--rw v4ur:next-hop-address?
      |                    +--:(v4ur:special-next-hop)
      |                    |  +--rw v4ur:special-next-hop?   enumeration
      |                    +--:(v4ur:next-hop-list)
      |                       +--rw v4ur:next-hop-list
      |                          +--rw v4ur:next-hop* [index]
      |                             +--rw v4ur:index              string
      |                             +--rw v4ur:outgoing-interface?
      |                             +--rw v4ur:next-hop-address?
      |
      +--rw ribs
         +--rw rib* [name]
            +--rw name               string
            +--rw address-family?    identityref
            +--rw description?       string
```

A.2. State Data

```
+--ro routing-state
|  +--ro router-id?                yang:dotted-quad
|  +--ro interfaces
|  |  +--ro interface*   if:interface-state-ref
|  +--ro control-plane-protocols
|  |  +--ro control-plane-protocol* [type name]
|  |     +--ro type    identityref
|  |     +--ro name    string
|  +--ro ribs
|     +--ro rib* [name]
|        +--ro name              string
|        +--ro address-family    identityref
|        +--ro default-rib?      boolean {multiple-ribs}?
|        +--ro routes
|        |  +--ro route*
|        |     +--ro route-preference?          route-preference
|        |     +--ro next-hop
|        |     |  +--ro (next-hop-options)
|        |     |     +--:(simple-next-hop)
|        |     |     |  +--ro outgoing-interface?
|        |     |     |  +--ro v6ur:next-hop-address?
|        |     |     |  +--ro v4ur:next-hop-address?
|        |     |     +--:(special-next-hop)
|        |     |     |  +--ro special-next-hop?        enumeration
|        |     |     +--:(next-hop-list)
|        |     |        +--ro next-hop-list
|        |     |           +--ro next-hop*
|        |     |              +--ro outgoing-interface?
|        |     |              +--ro v6ur:address?
|        |     |              +--ro v4ur:address?
|        |     +--ro source-protocol            identityref
|        |     +--ro active?                    empty
|        |     +--ro last-updated?              yang:date-and-time
|        |     +--ro v6ur:destination-prefix?   inet:ipv6-prefix
|        |     +--ro v4ur:destination-prefix?   inet:ipv4-prefix
|        +---x active-route
|           +---w input
|           |  +---w v6ur:destination-address?   inet:ipv6-address
|           |  +---w v4ur:destination-address?   inet:ipv4-address
|           +--ro output
|              +--ro route
|                 +--ro next-hop
|                 |  +--ro (next-hop-options)
|                 |     +--:(simple-next-hop)
|                 |     |  +--ro outgoing-interface?
|                 |     |  +--ro v6ur:next-hop-address?
|
```

```
|                  |     |  +--ro v4ur:next-hop-address?
|                  |     +--:(special-next-hop)
|                  |     |  +--ro special-next-hop?     enumeration
|                  |     +--:(next-hop-list)
|                  |        +--ro next-hop-list
|                  |           +--ro next-hop*
|                  |              +--ro outgoing-interface?
|                  |              +--ro v6ur:next-hop-address?
|                  |              +--ro v4ur:next-hop-address?
|                  +--ro source-protocol            identityref
|                  +--ro active?                    empty
|                  +--ro last-updated?           yang:date-and-time
|                  +--ro v6ur:destination-prefix?  inet:ipv6-prefix
|                  +--ro v4ur:destination-prefix?  inet:ipv4-prefix
```

Appendix B. Minimum Implementation

Some parts and options of the core routing model, such as user-defined RIBs, are intended only for advanced routers. This appendix gives basic non-normative guidelines for implementing a bare minimum of available functions. Such an implementation may be used for hosts or very simple routers.

A minimum implementation does not support the feature "multiple-ribs". This means that a single system-controlled RIB is available for each supported address family - IPv4, IPv6 or both. These RIBs are also the default RIBs. No user-controlled RIBs are allowed.

In addition to the mandatory instance of the "direct" pseudo-protocol, a minimum implementation should support configuring instance(s) of the "static" pseudo-protocol.

For hosts that are never intended to act as routers, the ability to turn on sending IPv6 router advertisements (Section 5.4) should be removed.

Platforms with severely constrained resources may use deviations for restricting the data model, e.g., limiting the number of "static" control plane protocol instances.

Appendix C. Example: Adding a New Control Plane Protocol

This appendix demonstrates how the core routing data model can be extended to support a new control plane protocol. The YANG module "example-rip" shown below is intended as an illustration rather than a real definition of a data model for the RIP routing protocol. For the sake of brevity, this module does not obey all the guidelines specified in [RFC6087]. See also Section 5.3.2.

```
   module example-rip {

     yang-version "1.1";

     namespace "http://example.com/rip";

     prefix "rip";

     import ietf-interfaces {
       prefix "if";
     }

     import ietf-routing {
       prefix "rt";
     }

     identity rip {
       base rt:routing-protocol;
       description
         "Identity for the RIP routing protocol.";
     }

     typedef rip-metric {
       type uint8 {
         range "0..16";
       }
     }

     grouping route-content {
       description
         "This grouping defines RIP-specific route attributes.";
       leaf metric {
         type rip-metric;
       }
       leaf tag {
         type uint16;
         default "0";
         description
           "This leaf may be used to carry additional info, e.g. AS
            number.";
       }
     }

     augment "/rt:routing-state/rt:ribs/rt:rib/rt:routes/rt:route" {
       when "derived-from-or-self(rt:source-protocol, 'rip:rip')" {
         description
           "This augment is only valid for a routes whose source
            protocol is RIP.";
```

```
      }
      description
        "RIP-specific route attributes.";
      uses route-content;
    }

    augment "/rt:routing-state/rt:ribs/rt:rib/rt:active-route/"
          + "rt:output/rt:route" {
      description
        "RIP-specific route attributes in the output of 'active-route'
         RPC.";
      uses route-content;
    }

    augment "/rt:routing/rt:control-plane-protocols/"
          + "rt:control-plane-protocol" {
      when "derived-from-or-self(rt:type,'rip:rip')" {
        description
          "This augment is only valid for a routing protocol instance
           of type 'rip'.";
      }
      container rip {
        presence "RIP configuration";
        description
          "RIP instance configuration.";
        container interfaces {
          description
            "Per-interface RIP configuration.";
          list interface {
            key "name";
            description
              "RIP is enabled on interfaces that have an entry in this
               list, unless 'enabled' is set to 'false' for that
               entry.";
            leaf name {
              type if:interface-ref;
            }
            leaf enabled {
              type boolean;
              default "true";
            }
            leaf metric {
              type rip-metric;
              default "1";
            }
          }
        }
        leaf update-interval {
```

```
          type uint8 {
            range "10..60";
          }
          units "seconds";
          default "30";
          description
            "Time interval between periodic updates.";
        }
      }
    }
  }
```

Appendix D. Data Tree Example

This section contains an example instance data tree in the JSON encoding [RFC7951], containing both configuration and state data. The data conforms to a data model that is defined by the following YANG library specification [RFC7895]:

```
{
  "ietf-yang-library:modules-state": {
    "module-set-id": "c2e1f54169aa7f36e1a6e8d0865d441d3600f9c4",
    "module": [
      {
        "name": "ietf-routing",
        "revision": "2016-11-03",
        "feature": [
          "multiple-ribs",
          "router-id"
        ],
        "namespace": "urn:ietf:params:xml:ns:yang:ietf-routing",
        "conformance-type": "implement"
      },
      {
        "name": "ietf-ipv4-unicast-routing",
        "revision": "2016-11-03",
        "namespace":
          "urn:ietf:params:xml:ns:yang:ietf-ipv4-unicast-routing",
        "conformance-type": "implement"
      },
      {
        "name": "ietf-ipv6-unicast-routing",
        "revision": "2016-11-03",
        "namespace":
          "urn:ietf:params:xml:ns:yang:ietf-ipv6-unicast-routing",
        "conformance-type": "implement"
      },
      {
```

```
          "name": "ietf-interfaces",
          "revision": "2014-05-08",
          "namespace": "urn:ietf:params:xml:ns:yang:ietf-interfaces",
          "conformance-type": "implement"
        },
        {
          "name": "ietf-inet-types",
          "namespace": "urn:ietf:params:xml:ns:yang:ietf-inet-types",
          "revision": "2013-07-15",
          "conformance-type": "import"
        },
        {
          "name": "ietf-yang-types",
          "namespace": "urn:ietf:params:xml:ns:yang:ietf-yang-types",
          "revision": "2013-07-15",
          "conformance-type": "import"
        },
        {
          "name": "iana-if-type",
          "namespace": "urn:ietf:params:xml:ns:yang:iana-if-type",
          "revision": "",
          "conformance-type": "implement"
        },
        {
          "name": "ietf-ip",
          "revision": "2014-06-16",
          "namespace": "urn:ietf:params:xml:ns:yang:ietf-ip",
          "conformance-type": "implement"
        }
      ]
    }
  }
```

A simple network set-up as shown in Figure 3 is assumed: router "A" uses static default routes with the "ISP" router as the next-hop. IPv6 router advertisements are configured only on the "eth1" interface and disabled on the upstream "eth0" interface.

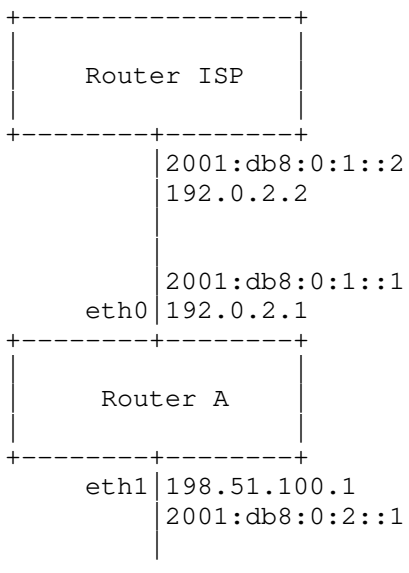

Figure 3: Example network configuration

The instance data tree could then be as follows:

```
{
  "ietf-interfaces:interfaces": {
    "interface": [
      {
        "name": "eth0",
        "type": "iana-if-type:ethernetCsmacd",
        "description": "Uplink to ISP.",
        "ietf-ip:ipv4": {
          "address": [
            {
              "ip": "192.0.2.1",
              "prefix-length": 24
            }
          ],
          "forwarding": true
        },
        "ietf-ip:ipv6": {
          "address": [
            {
              "ip": "2001:0db8:0:1::1",
              "prefix-length": 64
            }
          ],
          "forwarding": true,
```

```
            "autoconf": {
              "create-global-addresses": false
            }
          }
        },
        {
          "name": "eth1",
          "type": "iana-if-type:ethernetCsmacd",
          "description": "Interface to the internal network.",
          "ietf-ip:ipv4": {
            "address": [
              {
                "ip": "198.51.100.1",
                "prefix-length": 24
              }
            ],
            "forwarding": true
          },
          "ietf-ip:ipv6": {
            "address": [
              {
                "ip": "2001:0db8:0:2::1",
                "prefix-length": 64
              }
            ],
            "forwarding": true,
            "autoconf": {
              "create-global-addresses": false
            },
            "ietf-ipv6-unicast-routing:ipv6-router-advertisements": {
              "send-advertisements": true
            }
          }
        }
      ]
    },
    "ietf-interfaces:interfaces-state": {
      "interface": [
        {
          "name": "eth0",
          "type": "iana-if-type:ethernetCsmacd",
          "phys-address": "00:0C:42:E5:B1:E9",
          "oper-status": "up",
          "statistics": {
            "discontinuity-time": "2015-10-24T17:11:27+02:00"
          },
          "ietf-ip:ipv4": {
            "forwarding": true,
```

```
            "mtu": 1500,
            "address": [
              {
                "ip": "192.0.2.1",
                "prefix-length": 24
              }
            ]
          },
          "ietf-ip:ipv6": {
            "forwarding": true,
            "mtu": 1500,
            "address": [
              {
                "ip": "2001:0db8:0:1::1",
                "prefix-length": 64
              }
            ],
            "ietf-ipv6-unicast-routing:ipv6-router-advertisements": {
              "send-advertisements": true,
              "prefix-list": {
                "prefix": [
                  {
                    "prefix-spec": "2001:db8:0:2::/64"
                  }
                ]
              }
            }
          }
        },
        {
          "name": "eth1",
          "type": "iana-if-type:ethernetCsmacd",
          "phys-address": "00:0C:42:E5:B1:EA",
          "oper-status": "up",
          "statistics": {
            "discontinuity-time": "2015-10-24T17:11:29+02:00"
          },
          "ietf-ip:ipv4": {
            "forwarding": true,
            "mtu": 1500,
            "address": [
              {
                "ip": "198.51.100.1",
                "prefix-length": 24
              }
            ]
          },
          "ietf-ip:ipv6": {
```

```
            "forwarding": true,
            "mtu": 1500,
            "address": [
              {
                "ip": "2001:0db8:0:2::1",
                "prefix-length": 64
              }
            ],
            "ietf-ipv6-unicast-routing:ipv6-router-advertisements": {
              "send-advertisements": true,
              "prefix-list": {
                "prefix": [
                  {
                    "prefix-spec": "2001:db8:0:2::/64"
                  }
                ]
              }
            }
          }
        }
      ]
    },
    "ietf-routing:routing": {
      "router-id": "192.0.2.1",
      "control-plane-protocols": {
        "control-plane-protocol": [
          {
            "type": "ietf-routing:static",
            "name": "st0",
            "description":
              "Static routing is used for the internal network.",
            "static-routes": {
              "ietf-ipv4-unicast-routing:ipv4": {
                "route": [
                  {
                    "destination-prefix": "0.0.0.0/0",
                    "next-hop": {
                      "next-hop-address": "192.0.2.2"
                    }
                  }
                ]
              },
              "ietf-ipv6-unicast-routing:ipv6": {
                "route": [
                  {
                    "destination-prefix": "::/0",
                    "next-hop": {
                      "next-hop-address": "2001:db8:0:1::2"
```

```
                    }
                  }
                ]
              }
            }
          }
        ]
      }
    },
    "ietf-routing:routing-state": {
      "interfaces": {
        "interface": [
          "eth0",
          "eth1"
        ]
      },
      "control-plane-protocols": {
        "control-plane-protocol": [
          {
            "type": "ietf-routing:static",
            "name": "st0"
          }
        ]
      },
      "ribs": {
        "rib": [
          {
            "name": "ipv4-master",
            "address-family":
              "ietf-ipv4-unicast-routing:ipv4-unicast",
            "default-rib": true,
            "routes": {
              "route": [
                {
                  "ietf-ipv4-unicast-routing:destination-prefix":
                    "192.0.2.1/24",
                  "next-hop": {
                    "outgoing-interface": "eth0"
                  },
                  "route-preference": 0,
                  "source-protocol": "ietf-routing:direct",
                  "last-updated": "2015-10-24T17:11:27+02:00"
                },
                {
                  "ietf-ipv4-unicast-routing:destination-prefix":
                    "198.51.100.0/24",
                  "next-hop": {
                    "outgoing-interface": "eth1"
```

```
                  },
                  "source-protocol": "ietf-routing:direct",
                  "route-preference": 0,
                  "last-updated": "2015-10-24T17:11:27+02:00"
                },
                {
                  "ietf-ipv4-unicast-routing:destination-prefix":
                    "0.0.0.0/0",
                  "source-protocol": "ietf-routing:static",
                  "route-preference": 5,
                  "next-hop": {
                    "ietf-ipv4-unicast-routing:next-hop-address":
                      "192.0.2.2"
                  },
                  "last-updated": "2015-10-24T18:02:45+02:00"
                }
              ]
            }
          },
          {
            "name": "ipv6-master",
            "address-family":
              "ietf-ipv6-unicast-routing:ipv6-unicast",
            "default-rib": true,
            "routes": {
              "route": [
                {
                  "ietf-ipv6-unicast-routing:destination-prefix":
                    "2001:db8:0:1::/64",
                  "next-hop": {
                    "outgoing-interface": "eth0"
                  },
                  "source-protocol": "ietf-routing:direct",
                  "route-preference": 0,
                  "last-updated": "2015-10-24T17:11:27+02:00"
                },
                {
                  "ietf-ipv6-unicast-routing:destination-prefix":
                    "2001:db8:0:2::/64",
                  "next-hop": {
                    "outgoing-interface": "eth1"
                  },
                  "source-protocol": "ietf-routing:direct",
                  "route-preference": 0,
                  "last-updated": "2015-10-24T17:11:27+02:00"
                },
                {
                  "ietf-ipv6-unicast-routing:destination-prefix":
```

```
                    "::/0",
                  "next-hop": {
                    "ietf-ipv6-unicast-routing:next-hop-address":
                      "2001:db8:0:1::2"
                  },
                  "source-protocol": "ietf-routing:static",
                  "route-preference": 5,
                  "last-updated": "2015-10-24T18:02:45+02:00"
                }
              ]
            }
          }
        ]
      }
    }
  }
```

Appendix E. Change Log

RFC Editor: Remove this section upon publication as an RFC.

E.1. Changes Between Versions -24 and -25

- Minor edits based on IETF Last Call reviews.

E.2. Changes Between Versions -23 and -24

- Fix paths in "when" expressions due to errata 4749 of RFC 7950.

E.3. Changes Between Versions -22 and -23

- Removed "route-tag" feature.
- Removed next-hop classifiers.
- Fixed invalid when expressions in augments.
- In simple-next-hop, an address, outgoing interface or both can be specified.
- RPC "fib-route" changed into RIB action "active-route".
- The requirement that direct routes be always placed in default RIBs.

E.4. Changes Between Versions -21 and -22

- Added "next-hop-list" as a new case of the "next-hop-options" choice.
- Renamed "routing protocol" to "control plane protocol" in both the YANG modules and I-D text.

E.5. Changes Between Versions -20 and -21

- Routing instances were removed.
- IPv6 RA parameters were moved to the "ietf-ipv6-router-advertisements".

E.6. Changes Between Versions -19 and -20

- Assignment of L3 interfaces to routing instances is now part of interface configuration.
- Next-hop options in configuration were aligned with state data.
- It is recommended to enclose protocol-specific configuration in a presence container.

E.7. Changes Between Versions -18 and -19

- The leaf "route-preference" was removed from the "routing-protocol" container in both "routing" and "routing-state".
- The "vrf-routing-instance" identity was added in support of a common routing-instance type in addition to the "default-routing-instance".
- Removed "enabled" switch from "routing-protocol".

E.8. Changes Between Versions -17 and -18

- The container "ribs" was moved under "routing-instance" (in both "routing" and "routing-state").
- Typedefs "rib-ref" and "rib-state-ref" were removed.
- Removed "recipient-ribs" (both state and configuration).
- Removed "connected-ribs" from "routing-protocol" (both state and configuration).

- Configuration and state data for IPv6 RA were moved under "if:interface" and "if:interface-state".

- Assignment of interfaces to routing instances now use leaf-list rather than list (both config and state). The opposite reference from "if:interface" to "rt:routing-instance" was changed to a single leaf (an interface cannot belong to multiple routing instances).

- Specification of a default RIB is now a simple flag under "rib" (both config and state).

- Default RIBs are marked by a flag in state data.

E.9. Changes Between Versions -16 and -17

- Added Acee as a co-author.

- Removed all traces of route filters.

- Removed numeric IDs of list entries in state data.

- Removed all next-hop cases except "simple-next-hop" and "special-next-hop".

- Removed feature "multipath-routes".

- Augmented "ietf-interfaces" module with a leaf-list of leafrefs pointing form state data of an interface entry to the routing instance(s) to which the interface is assigned.

E.10. Changes Between Versions -15 and -16

- Added 'type' as the second key component of 'routing-protocol', both in configuration and state data.

- The restriction of no more than one connected RIB per address family was removed.

- Removed the 'id' key of routes in RIBs. This list has no keys anymore.

- Remove the 'id' key from static routes and make 'destination-prefix' the only key.

- Added 'route-preference' as a new attribute of routes in RIB.

- Added 'active' as a new attribute of routes in RIBs.

- Renamed RPC operation 'active-route' to 'fib-route'.
- Added 'route-preference' as a new parameter of routing protocol instances, both in configuration and state data.
- Renamed identity 'rt:standard-routing-instance' to 'rt:default-routing-instance'.
- Added next-hop lists to state data.
- Added two cases for specifying next-hops indirectly - via a new RIB or a recursive list of next-hops.
- Reorganized next-hop in static routes.
- Removed all 'if-feature' statements from state data.

E.11. Changes Between Versions -14 and -15

- Removed all defaults from state data.
- Removed default from 'cur-hop-limit' in config.

E.12. Changes Between Versions -13 and -14

- Removed dependency of 'connected-ribs' on the 'multiple-ribs' feature.
- Removed default value of 'cur-hop-limit' in state data.
- Moved parts of descriptions and all references on IPv6 RA parameters from state data to configuration.
- Added reference to RFC 6536 in the Security section.

E.13. Changes Between Versions -12 and -13

- Wrote appendix about minimum implementation.
- Remove "when" statement for IPv6 router interface state data - it was dependent on a config value that may not be present.
- Extra container for the next-hop list.
- Names rather than numeric ids are used for referring to list entries in state data.

- Numeric ids are always declared as mandatory and unique. Their description states that they are ephemeral.
- Descriptions of "name" keys in state data lists are required to be persistent.
-
- Removed "if-feature multiple-ribs;" from connected-ribs.
- "rib-name" instead of "name" is used as the name of leafref nodes.
- "next-hop" instead of "nexthop" or "gateway" used throughout, both in node names and text.

E.14. Changes Between Versions -11 and -12

- Removed feature "advanced-router" and introduced two features instead: "multiple-ribs" and "multipath-routes".
- Unified the keys of config and state versions of "routing-instance" and "rib" lists.
- Numerical identifiers of state list entries are not keys anymore, but they are constrained using the "unique" statement.
- Updated acknowledgements.

E.15. Changes Between Versions -10 and -11

- Migrated address families from IANA enumerations to identities.
- Terminology and node names aligned with the I2RS RIB model: router -> routing instance, routing table -> RIB.
- Introduced uint64 keys for state lists: routing-instance, rib, route, nexthop.
- Described the relationship between system-controlled and user-controlled list entries.
- Feature "user-defined-routing-tables" changed into "advanced-router".
- Made nexthop into a choice in order to allow for nexthop-list (I2RS requirement).

- Added nexthop-list with entries having priorities (backup) and weights (load balancing).

- Updated bibliography references.

E.16. Changes Between Versions -09 and -10

- Added subtree for state data ("/routing-state").

- Terms "system-controlled entry" and "user-controlled entry" defined and used.

- New feature "user-defined-routing-tables". Nodes that are useful only with user-defined routing tables are now conditional.

- Added grouping "router-id".

- In routing tables, "source-protocol" attribute of routes now reports only protocol type, and its datatype is "identityref".

- Renamed "main-routing-table" to "default-routing-table".

E.17. Changes Between Versions -08 and -09

- Fixed "must" expression for "connected-routing-table".

- Simplified "must" expression for "main-routing-table".

- Moved per-interface configuration of a new routing protocol under 'routing-protocol'. This also affects the 'example-rip' module.

E.18. Changes Between Versions -07 and -08

- Changed reference from RFC6021 to RFC6021bis.

E.19. Changes Between Versions -06 and -07

- The contents of <get-reply> in Appendix D was updated: "eth[01]" is used as the value of "location", and "forwarding" is on for both interfaces and both IPv4 and IPv6.

- The "must" expression for "main-routing-table" was modified to avoid redundant error messages reporting address family mismatch when "name" points to a non-existent routing table.

- The default behavior for IPv6 RA prefix advertisements was clarified.

- Changed type of "rt:router-id" to "ip:dotted-quad".
- Type of "rt:router-id" changed to "yang:dotted-quad".
- Fixed missing prefixes in XPath expressions.

E.20. Changes Between Versions -05 and -06

- Document title changed: "Configuration" was replaced by "Management".
- New typedefs "routing-table-ref" and "route-filter-ref".
- Double slashes "//" were removed from XPath expressions and replaced with the single "/".
- Removed uniqueness requirement for "router-id".
- Complete data tree is now in Appendix A.
- Changed type of "source-protocol" from "leafref" to "string".
- Clarified the relationship between routing protocol instances and connected routing tables.
- Added a must constraint saying that a routing table connected to the direct pseudo-protocol must not be a main routing table.

E.21. Changes Between Versions -04 and -05

- Routing tables are now global, i.e., "routing-tables" is a child of "routing" rather than "router".
- "must" statement for "static-routes" changed to "when".
- Added "main-routing-tables" containing references to main routing tables for each address family.
- Removed the defaults for "address-family" and "safi" and made them mandatory.
- Removed the default for route-filter/type and made this leaf mandatory.
- If there is no active route for a given destination, the "active-route" RPC returns no output.
- Added "enabled" switch under "routing-protocol".

- Added "router-type" identity and "type" leaf under "router".
- Route attribute "age" changed to "last-updated", its type is "yang:date-and-time".
- The "direct" pseudo-protocol is always connected to main routing tables.
- Entries in the list of connected routing tables renamed from "routing-table" to "connected-routing-table".
- Added "must" constraint saying that a routing table must not be its own recipient.

E.22. Changes Between Versions -03 and -04

- Changed "error-tag" for both RPC operations from "missing element" to "data-missing".
- Removed the decrementing behavior for advertised IPv6 prefix parameters "valid-lifetime" and "preferred-lifetime".
- Changed the key of the static route lists from "seqno" to "id" because the routes needn't be sorted.
- Added 'must' constraint saying that "preferred-lifetime" must not be greater than "valid-lifetime".

E.23. Changes Between Versions -02 and -03

- Module "iana-afn-safi" moved to I-D "iana-if-type".
- Removed forwarding table.
- RPC "get-route" changed to "active-route". Its output is a list of routes (for multi-path routing).
- New RPC "route-count".
- For both RPCs, specification of negative responses was added.
- Relaxed separation of router instances.
- Assignment of interfaces to router instances needn't be disjoint.
- Route filters are now global.
- Added "allow-all-route-filter" for symmetry.

- Added Section 6 about interactions with "ietf-interfaces" and "ietf-ip".
- Added "router-id" leaf.
- Specified the names for IPv4/IPv6 unicast main routing tables.
- Route parameter "last-modified" changed to "age".
- Added container "recipient-routing-tables".

E.24. Changes Between Versions -01 and -02

- Added module "ietf-ipv6-unicast-routing".
- The example in Appendix D now uses IP addresses from blocks reserved for documentation.
- Direct routes appear by default in the forwarding table.
- Network layer interfaces must be assigned to a router instance. Additional interface configuration may be present.
- The "when" statement is only used with "augment", "must" is used elsewhere.
- Additional "must" statements were added.
- The "route-content" grouping for IPv4 and IPv6 unicast now includes the material from the "ietf-routing" version via "uses rt:route-content".
- Explanation of symbols in the tree representation of data model hierarchy.

E.25. Changes Between Versions -00 and -01

- AFN/SAFI-independent stuff was moved to the "ietf-routing" module.
- Typedefs for AFN and SAFI were placed in a separate "iana-afn-safi" module.
- Names of some data nodes were changed, in particular "routing-process" is now "router".
- The restriction of a single AFN/SAFI per router was lifted.
- RPC operation "delete-route" was removed.

- Illegal XPath references from "get-route" to the datastore were fixed.
- Section "Security Considerations" was written.

Authors' Addresses

Ladislav Lhotka
CZ.NIC

Email: lhotka@nic.cz

Acee Lindem
Cisco Systems

Email: acee@cisco.com

Routing Area Working Group
Internet-Draft
Intended status: Informational
Expires: December 6, 2016

P. Lapukhov
Facebook
A. Premji
Arista Networks
J. Mitchell, Ed.
June 4, 2016

Use of BGP for routing in large-scale data centers
draft-ietf-rtgwg-bgp-routing-large-dc-11

Abstract

Some network operators build and operate data centers that support over one hundred thousand servers. In this document, such data centers are referred to as "large-scale" to differentiate them from smaller infrastructures. Environments of this scale have a unique set of network requirements with an emphasis on operational simplicity and network stability. This document summarizes operational experience in designing and operating large-scale data centers using BGP as the only routing protocol. The intent is to report on a proven and stable routing design that could be leveraged by others in the industry.

Table of Contents

1. Introduction

This document describes a practical routing design that can be used in a large-scale data center (DC) design. Such data centers, also known as hyper-scale or warehouse-scale data centers, have a unique attribute of supporting over a hundred thousand servers. In order to accommodate networks of this scale, operators are revisiting networking designs and platforms to address this need.

The design presented in this document is based on operational experience with data centers built to support large-scale distributed software infrastructure, such as a Web search engine. The primary requirements in such an environment are operational simplicity and network stability so that a small group of people can effectively support a significantly sized network.

Experimentation and extensive testing have shown that External BGP (EBGP) [RFC4271] is well suited as a stand-alone routing protocol for these type of data center applications. This is in contrast with more traditional DC designs, which may use simple tree topologies and rely on extending Layer 2 domains across multiple network devices. This document elaborates on the requirements that led to this design choice and presents details of the EBGP routing design as well as explores ideas for further enhancements.

This document first presents an overview of network design requirements and considerations for large-scale data centers. Then traditional hierarchical data center network topologies are contrasted with Clos networks [CLOS1953] that are horizontally scaled out. This is followed by arguments for selecting EBGP with a Clos topology as the most appropriate routing protocol to meet the requirements and the proposed design is described in detail. Finally, this document reviews some additional considerations and design options. A thorough understanding of BGP is assumed by a

reader planning on deploying the design described within the document.

2. Network Design Requirements

This section describes and summarizes network design requirements for large-scale data centers.

2.1. Bandwidth and Traffic Patterns

The primary requirement when building an interconnection network for a large number of servers is to accommodate application bandwidth and latency requirements. Until recently it was quite common to see the majority of traffic entering and leaving the data center, commonly referred to as "north-south" traffic. Traditional "tree" topologies were sufficient to accommodate such flows, even with high oversubscription ratios between the layers of the network. If more bandwidth was required, it was added by "scaling up" the network elements, e.g., by upgrading the device's linecards or fabrics or replacing the device with one with higher port density.

Today many large-scale data centers host applications generating significant amounts of server-to-server traffic, which does not egress the DC, commonly referred to as "east-west" traffic. Examples of such applications could be compute clusters such as Hadoop [HADOOP], massive data replication between clusters needed by certain applications, or virtual machine migrations. Scaling traditional tree topologies to match these bandwidth demands becomes either too expensive or impossible due to physical limitations, e.g., port density in a switch.

2.2. CAPEX Minimization

The Capital Expenditures (CAPEX) associated with the network infrastructure alone constitutes about 10-15% of total data center expenditure (see [GREENBERG2009]). However, the absolute cost is significant, and hence there is a need to constantly drive down the cost of individual network elements. This can be accomplished in two ways:

- Unifying all network elements, preferably using the same hardware type or even the same device. This allows for volume pricing on bulk purchases and reduced maintenance and inventory costs.

- Driving costs down using competitive pressures, by introducing multiple network equipment vendors.

In order to allow for good vendor diversity it is important to minimize the software feature requirements for the network elements. This strategy provides maximum flexibility of vendor equipment choices while enforcing interoperability using open standards.

2.3. OPEX Minimization

Operating large-scale infrastructure can be expensive as a larger amount of elements will statistically fail more often. Having a simpler design and operating using a limited software feature set minimizes software issue-related failures.

An important aspect of Operational Expenditure (OPEX) minimization is reducing the size of failure domains in the network. Ethernet networks are known to be susceptible to broadcast or unicast traffic storms that can have a dramatic impact on network performance and availability. The use of a fully routed design significantly reduces the size of the data plane failure domains, i.e., limits them to the lowest level in the network hierarchy. However, such designs introduce the problem of distributed control plane failures. This observation calls for simpler and less control plane protocols to reduce protocol interaction issues, reducing the chance of a network meltdown. Minimizing software feature requirements as described in the CAPEX section above also reduces testing and training requirements.

2.4. Traffic Engineering

In any data center, application load balancing is a critical function performed by network devices. Traditionally, load balancers are deployed as dedicated devices in the traffic forwarding path. The problem arises in scaling load balancers under growing traffic demand. A preferable solution would be able to scale the load balancing layer horizontally, by adding more of the uniform nodes and distributing incoming traffic across these nodes. In situations like this, an ideal choice would be to use network infrastructure itself to distribute traffic across a group of load balancers. The combination of Anycast prefix advertisement [RFC4786] and Equal Cost Multipath (ECMP) functionality can be used to accomplish this goal. To allow for more granular load distribution, it is beneficial for the network to support the ability to perform controlled per-hop traffic engineering. For example, it is beneficial to directly control the ECMP next-hop set for Anycast prefixes at every level of network hierarchy.

2.5. Summarized Requirements

This section summarizes the list of requirements outlined in the previous sections:

- REQ1: Select a topology that can be scaled "horizontally" by adding more links and network devices of the same type without requiring upgrades to the network elements themselves.
- REQ2: Define a narrow set of software features/protocols supported by a multitude of networking equipment vendors.
- REQ3: Choose a routing protocol that has a simple implementation in terms of programming code complexity and ease of operational support.
- REQ4: Minimize the failure domain of equipment or protocol issues as much as possible.
- REQ5: Allow for some traffic engineering, preferably via explicit control of the routing prefix next-hop using built-in protocol mechanics.

3. Data Center Topologies Overview

This section provides an overview of two general types of data center designs - hierarchical (also known as tree based) and Clos based network designs.

3.1. Traditional DC Topology

In the networking industry, a common design choice for data centers typically look like an (upside down) tree with redundant uplinks and three layers of hierarchy namely; core, aggregation/distribution and access layers (see Figure 1). To accommodate bandwidth demands, each higher layer, from server towards DC egress or WAN, has higher port density and bandwidth capacity where the core functions as the "trunk" of the tree based design. To keep terminology uniform and for comparison with other designs, in this document these layers will be referred to as Tier-1, Tier-2 and Tier-3 "tiers", instead of Core, Aggregation or Access layers.

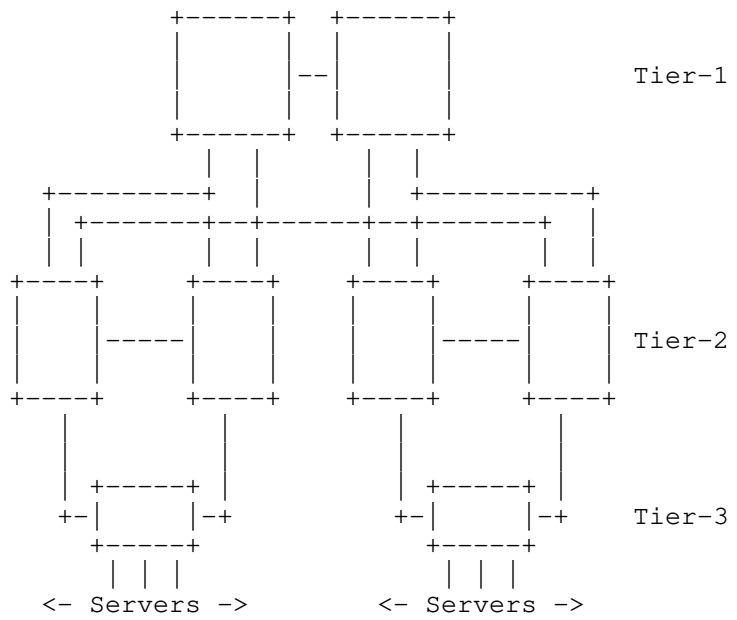

Figure 1: Typical DC network topology

Unfortunately, as noted previously, it is not possible to scale a tree based design to a large enough degree to handle large-scale designs due to the inability to be able to acquire Tier-1 devices with a large enough port density to sufficiently scale Tier-2. Also, continous upgrades or replacement of the upper tier devices are required as deployment size or bandwidth requirements increase which is operationally complex. For this reason, REQ1 is in place, eliminating this type of design from consideration.

3.2. Clos Network topology

This section describes a common design for horizontally scalable topology in large-scale data centers in order to meet REQ1.

3.2.1. Overview

A common choice for a horizontally scalable topology is a folded Clos topology, sometimes called "fat-tree" (see, for example, [INTERCON] and [ALFARES2008]). This topology features an odd number of stages (sometimes known as dimensions) and is commonly made of uniform elements, e.g., network switches with the same port count. Therefore, the choice of folded Clos topology satisfies REQ1 and facilitates REQ2. See Figure 2 below for an example of a folded

3-stage Clos topology (3 stages counting Tier-2 stage twice, when tracing a packet flow):

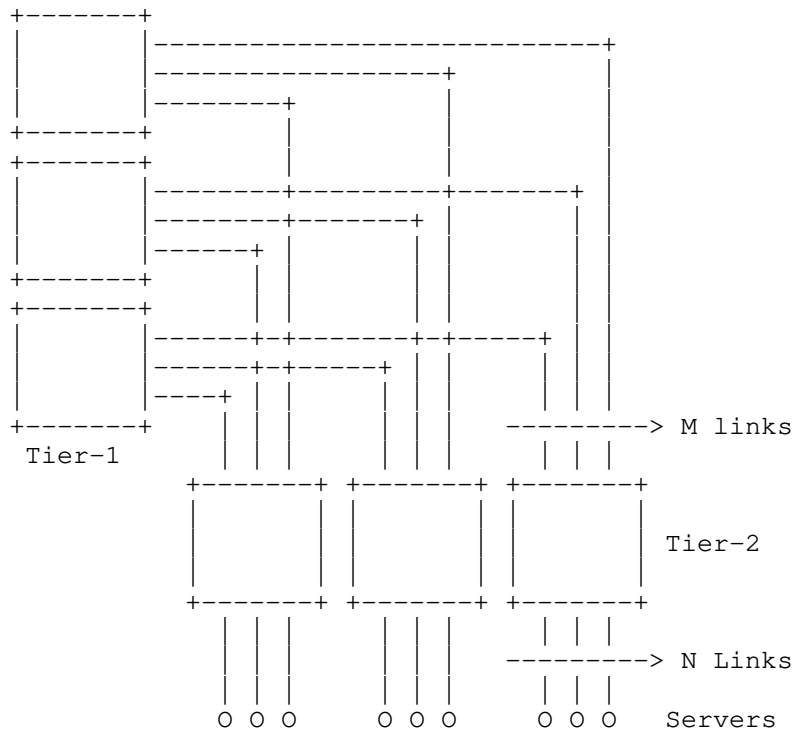

Figure 2: 3-Stage Folded Clos topology

This topology is often also referred to as a "Leaf and Spine" network, where "Spine" is the name given to the middle stage of the Clos topology (Tier-1) and "Leaf" is the name of input/output stage (Tier-2). For uniformity, this document will refer to these layers using the "Tier-n" notation.

3.2.2. Clos Topology Properties

The following are some key properties of the Clos topology:

- The topology is fully non-blocking, or more accurately non-interfering, if M >= N and oversubscribed by a factor of N/M otherwise. Here M and N is the uplink and downlink port count respectively, for a Tier-2 switch as shown in Figure 2.

- Utilizing this topology requires control and data plane support for ECMP with a fan-out of M or more.

- Tier-1 switches have exactly one path to every server in this topology. This is an important property that makes route summarization dangerous in this topology (see Section 8.2 below).

- Traffic flowing from server to server is load balanced over all available paths using ECMP.

3.2.3. Scaling the Clos topology

A Clos topology can be scaled either by increasing network element port density or adding more stages, e.g., moving to a 5-stage Clos, as illustrated in Figure 3 below:

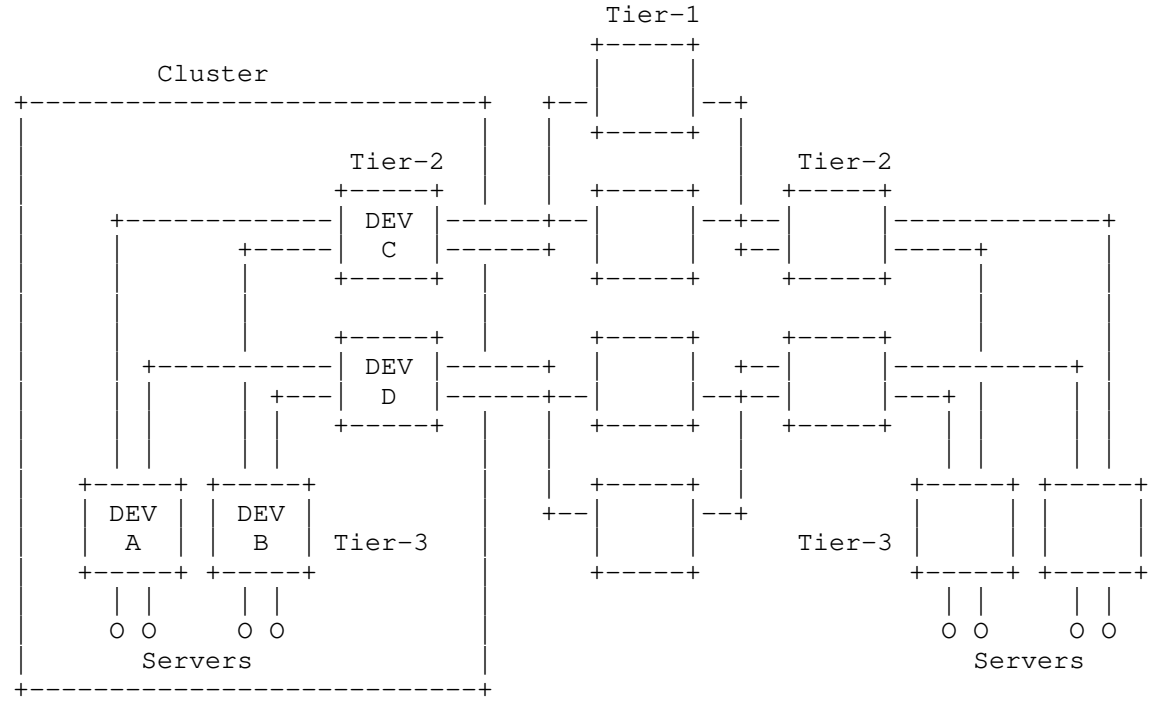

Figure 3: 5-Stage Clos topology

The small example topology on Figure 3 is built from devices with a port count of 4 and provides full bisectional bandwidth to all connected servers. In this document, one set of directly connected Tier-2 and Tier-3 devices along with their attached servers will be referred to as a "cluster". For example, DEV A, B, C, D, and the servers that connect to DEV A and B, on Figure 3 form a cluster. The concept of a cluster may also be a useful concept as a single

deployment or maintenance unit which can be operated on at a different frequency than the entire topology.

In practice, the Tier-3 layer of the network, which are typically top of rack switches (ToRs), is where oversubscription is introduced to allow for packaging of more servers in the data center while meeting the bandwidth requirements for different types of applications. The main reason to limit oversubscription at a single layer of the network is to simplify application development that would otherwise need to account for multiple bandwidth pools: within rack (Tier-3), between racks (Tier-2), and between clusters (Tier-1). Since oversubscription does not have a direct relationship to the routing design it is not discussed further in this document.

3.2.4. Managing the Size of Clos Topology Tiers

If a data center network size is small, it is possible to reduce the number of switches in Tier-1 or Tier-2 of a Clos topology by a factor of two. To understand how this could be done, take Tier-1 as an example. Every Tier-2 device connects to a single group of Tier-1 devices. If half of the ports on each of the Tier-1 devices are not being used then it is possible to reduce the number of Tier-1 devices by half and simply map two uplinks from a Tier-2 device to the same Tier-1 device that were previously mapped to different Tier-1 devices. This technique maintains the same bisectional bandwidth while reducing the number of elements in the Tier-1 layer, thus saving on CAPEX. The tradeoff, in this example, is the reduction of maximum DC size in terms of overall server count by half.

In this example, Tier-2 devices will be using two parallel links to connect to each Tier-1 device. If one of these links fails, the other will pick up all traffic of the failed link, possible resulting in heavy congestion and quality of service degradation if the path determination procedure does not take bandwidth amount into account since the number of upstream Tier-1 devices is likely wider than two. To avoid this situation, parallel links can be grouped in link aggregation groups (LAGs, such as [IEEE8023AD]) with widely available implementation settings that take the whole "bundle" down upon a single link failure. Equivalent techniques that enforce "fate sharing" on the parallel links can be used in place of LAGs to achieve the same effect. As a result of such fate-sharing, traffic from two or more failed links will be re-balanced over the multitude of remaining paths that equals the number of Tier-1 devices. This example is using two links for simplicity, having more links in a bundle will have less impact on capacity upon a member-link failure.

4. Data Center Routing Overview

This section provides an overview of three general types of data center protocol designs - Layer 2 only, Hybrid L2/L3 and Layer 3 only.

4.1. Layer 2 Only Designs

Originally most data center designs used Spanning-Tree Protocol (STP) originally defined in [IEEE8021D-1990] for loop free topology creation, typically utilizing variants of the traditional DC topology described in Section 3.1. At the time, many DC switches either did not support Layer 3 routing protocols or supported them with additional licensing fees, which played a part in the design choice. Although many enhancements have been made through the introduction of Rapid Spanning Tree Protocol (RSTP) in the latest revision of [IEEE8021D-2004] and Multiple Spanning Tree Protocol (MST) specified in [IEEE8021Q] that increase convergence, stability and load balancing in larger topologies, many of the fundamentals of the protocol limit its applicability in large-scale DCs. STP and its newer variants use an active/standby approach to path selection and are therefore hard to deploy in horizontally-scaled topologies as described in Section 3.2. Further, operators have had many experiences with large failures due to issues caused by improper cabling, misconfiguration, or flawed software on a single device. These failures regularly affected the entire spanning-tree domain and were very hard to troubleshoot due to the nature of the protocol. For these reasons, and since almost all DC traffic is now IP, therefore requiring a Layer 3 routing protocol at the network edge for external connectivity, designs utilizing STP usually fail all of the requirements of large-scale DC operators. Various enhancements to link-aggregation protocols such as [IEEE8023AD], generally known as Multi-Chassis Link-Aggregation (M-LAG) made it possible to use Layer 2 designs with active-active network paths while relying on STP as the backup for loop prevention. The major downsides of this approach are the lack of ability to scale linearly past two in most implementations, lack of standards based implementations, and added the failure domain risk of syncing state between the devices.

It should be noted that building large, horizontally scalable, Layer 2 only networks without STP is possible recently through the introduction of the TRILL protocol in [RFC6325]. TRILL resolves many of the issues STP has for large-scale DC design however due to the limited number of implementations, and often the requirement for specific equipment that supports it, this has limited its applicability and increased the cost of such designs.

Finally, neither the base TRILL specification nor the M-LAG approach totally eliminate the problem of the shared broadcast domain, that is so detrimental to the operations of any Layer 2, Ethernet based solution. Later TRILL extensions have been proposed to solve the this problem statement primarily based on the approaches outlined in [RFC7067], but this even further limits the number of available interoperable implementations that can be used to build a fabric. Therefore, TRILL based designs have issues meeting REQ2, REQ3, and REQ4.

4.2. Hybrid L2/L3 Designs

Operators have sought to limit the impact of data plane faults and build large-scale topologies through implementing routing protocols in either the Tier-1 or Tier-2 parts of the network and dividing the Layer 2 domain into numerous, smaller domains. This design has allowed data centers to scale up, but at the cost of complexity in managing multiple network protocols. For the following reasons, operators have retained Layer 2 in either the access (Tier-3) or both access and aggregation (Tier-3 and Tier-2) parts of the network:

- Supporting legacy applications that may require direct Layer 2 adjacency or use non-IP protocols.
- Seamless mobility for virtual machines that require the preservation of IP addresses when a virtual machine moves to a different Tier-3 switch.
- Simplified IP addressing = less IP subnets are required for the data center.
- Application load balancing may require direct Layer 2 reachability to perform certain functions such as Layer 2 Direct Server Return (DSR, see [L3DSR]).
- Continued CAPEX differences between Layer 2 and Layer 3 capable switches.

4.3. Layer 3 Only Designs

Network designs that leverage IP routing down to Tier-3 of the network have gained popularity as well. The main benefit of these designs is improved network stability and scalability, as a result of confining L2 broadcast domains. Commonly an Interior Gateway Protocol (IGP) such as OSPF [RFC2328] is used as the primary routing protocol in such a design. As data centers grow in scale, and server count exceeds tens of thousands, such fully routed designs have become more attractive.

Choosing a Layer 3 only design greatly simplifies the network, facilitating the meeting of REQ1 and REQ2, and has widespread adoption in networks where large Layer 2 adjacency and larger size Layer 3 subnets are not as critical compared to network scalability and stability. Application providers and network operators continue to develop new solutions to meet some of the requirements that previously had driven large Layer 2 domains by using various overlay or tunneling techniques.

5. Routing Protocol Design

In this section the motivations for using External BGP (EBGP) as the single routing protocol for data center networks having a Layer 3 protocol design and Clos topology are reviewed. Then, a practical approach for designing an EBGP based network is provided.

5.1. Choosing EBGP as the Routing Protocol

REQ2 would give preference to the selection of a single routing protocol to reduce complexity and interdependencies. While it is common to rely on an IGP in this situation, sometimes with either the addition of EBGP at the device bordering the WAN or Internal BGP (IBGP) throughout, this document proposes the use of an EBGP only design.

Although EBGP is the protocol used for almost all inter-domain routing in the Internet and has wide support from both vendor and service provider communities, it is not generally deployed as the primary routing protocol within the data center for a number of reasons (some of which are interrelated):

- BGP is perceived as a "WAN only protocol only" and not often considered for enterprise or data center applications.
- BGP is believed to have a "much slower" routing convergence compared to IGPs.
- Large scale BGP deployments typically utilize an IGP for BGP next-hop resolution as all nodes in the iBGP topology are not directly connected.
- BGP is perceived to require significant configuration overhead and does not support neighbor auto-discovery.

This document discusses some of these perceptions, especially as applicable to the proposed design, and highlights some of the advantages of using the protocol such as:

- BGP has less complexity in parts of its protocol design - internal data structures and state machine are simpler as compared to most link-state IGPs such as OSPF. For example, instead of implementing adjacency formation, adjacency maintenance and/or flow-control, BGP simply relies on TCP as the underlying transport. This fulfills REQ2 and REQ3.

- BGP information flooding overhead is less when compared to link-state IGPs. Since every BGP router calculates and propagates only the best-path selected, a network failure is masked as soon as the BGP speaker finds an alternate path, which exists when highly symmetric topologies, such as Clos, are coupled with an EBGP only design. In contrast, the event propagation scope of a link-state IGP is an entire area, regardless of the failure type. In this way, BGP better meets REQ3 and REQ4. It is also worth mentioning that all widely deployed link-state IGPs feature periodic refreshes of routing information while BGP does not expire routing state, although this rarely impacts modern router control planes.

- BGP supports third-party (recursively resolved) next-hops. This allows for manipulating multipath to be non-ECMP based or forwarding based on application-defined paths, through establishment of a peering session with an application "controller" which can inject routing information into the system, satisfying REQ5. OSPF provides similar functionality using concepts such as "Forwarding Address", but with more difficulty in implementation and far less control of information propagation scope.

- Using a well-defined Autonomous System Number (ASN) allocation scheme and standard AS_PATH loop detection, "BGP path hunting" (see [JAKMA2008]) can be controlled and complex unwanted paths will be ignored. See Section 5.2 for an example of a working ASN allocation scheme. In a link-state IGP accomplishing the same goal would require multi-(instance/topology/process) support, typically not available in all DC devices and quite complex to configure and troubleshoot. Using a traditional single flooding domain, which most DC designs utilize, under certain failure conditions may pick up unwanted lengthy paths, e.g., traversing multiple Tier-2 devices.

- EBGP configuration that is implemented with minimal routing policy is easier to troubleshoot for network reachability issues. In most implementations, it is straightforward to view contents of BGP Loc-RIB and compare it to the router's RIB. Also, in most implementations an operator can view every BGP neighbors Adj-RIB-In and Adj-RIB-Out structures and therefore incoming and outgoing

NLRI information can be easily correlated on both sides of a BGP session. Thus, BGP satisfies REQ3.

5.2. EBGP Configuration for Clos topology

Clos topologies that have more than 5 stages are very uncommon due to the large numbers of interconnects required by such a design. Therefore, the examples below are made with reference to the 5-stage Clos topology (in unfolded state).

5.2.1. EBGP Configuration Guidelines and Example ASN Scheme

The diagram below illustrates an example ASN allocation scheme. The following is a list of guidelines that can be used:

- EBGP single-hop sessions are established over direct point-to-point links interconnecting the network nodes, no multi-hop or loopback sessions are used even in the case of multiple links between the same pair of nodes.
- Private Use ASNs from the range 64512-65534 are used to avoid ASN conflicts.
- A single ASN is allocated to all of the Clos topology's Tier-1 devices.
- A unique ASN is allocated to each set of Tier-2 devices in the same cluster.
- A unique ASN is allocated to every Tier-3 device (e.g., ToR) in this topology.

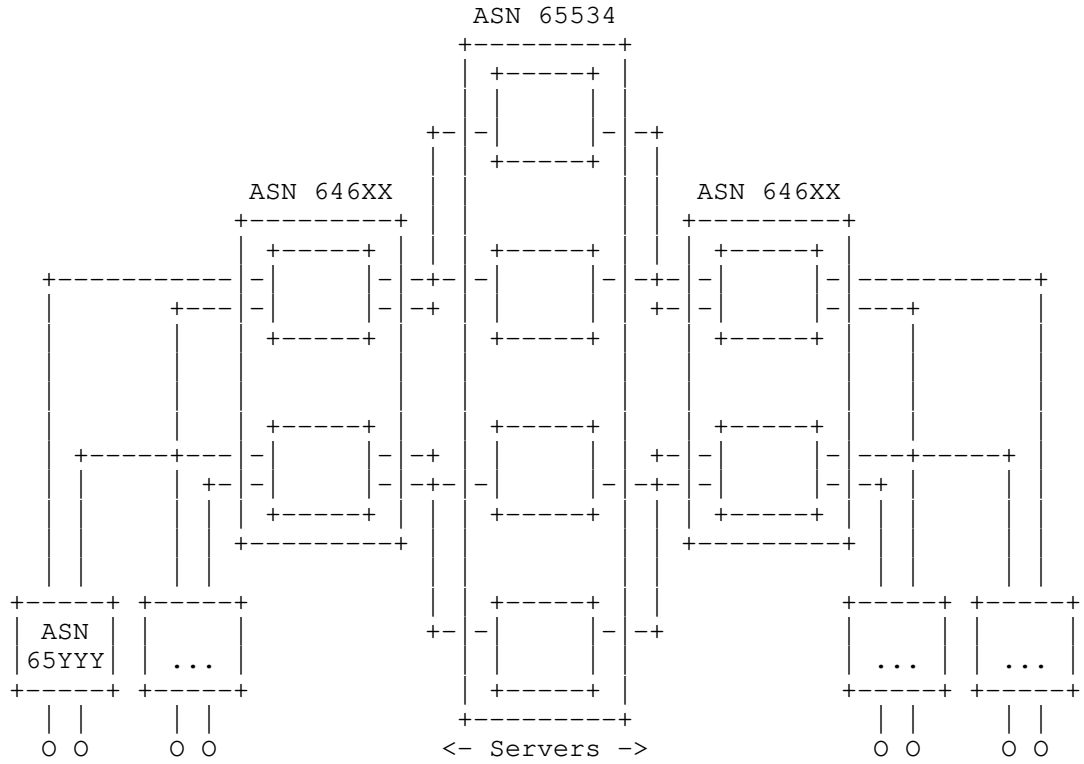

Figure 4: BGP ASN layout for 5-stage Clos

5.2.2. Private Use ASNs

The original range of Private Use ASNs [RFC6996] limited operators to 1023 unique ASNs. Since it is quite likely that the number of network devices may exceed this number, a workaround is required. One approach is to re-use the ASNs assigned to the Tier-3 devices across different clusters. For example, Private Use ASNs 65001, 65002 ... 65032 could be used within every individual cluster and assigned to Tier-3 devices.

To avoid route suppression due to the AS_PATH loop detection mechanism in BGP, upstream EBGP sessions on Tier-3 devices must be configured with the "AllowAS In" feature [ALLOWASIN] that allows accepting a device's own ASN in received route advertisements. Although this feature is not standarized, it is widely available accross multiple vendors implementations. Introducing this feature does not make routing loops more likely in the design since the AS_PATH is being added to by routers at each of the topology tiers and AS_PATH length is an early tie breaker in the BGP path selection

process. Further loop protection is still in place at the Tier-1 device, which will not accept routes with a path including its own ASN and Tier-2 devices do not have direct connectivity with each other.

Another solution to this problem would be using Four-Octet ASNs ([RFC6793]), where there are additional Private Use ASNs available, see [IANA.AS]. Use of Four-Octet ASNs puts additional protocol complexity in the BGP implementation and should be balanced against the complexity of re-use when considering REQ3 and REQ4. Perhaps more importantly, they are not yet supported by all BGP implementations, which may limit vendor selection of DC equipment. When supported, ensure that deployed implementations are able to remove the Private Use ASNs when external connectivity (Section 5.2.4) to these ASNs is required.

5.2.3. Prefix Advertisement

A Clos topology features a large number of point-to-point links and associated prefixes. Advertising all of these routes into BGP may create Forwarding Information Base (FIB) overload in the network devices. Advertising these links also puts additional path computation stress on the BGP control plane for little benefit. There are two possible solutions:

- Do not advertise any of the point-to-point links into BGP. Since the EBGP-based design changes the next-hop address at every device, distant networks will automatically be reachable via the advertising EBGP peer and do not require reachability to these prefixes. However, this may complicate operations or monitoring: e.g., using the popular "traceroute" tool will display IP addresses that are not reachable.

- Advertise point-to-point links, but summarize them on every device. This requires an address allocation scheme such as allocating a consecutive block of IP addresses per Tier-1 and Tier-2 device to be used for point-to-point interface addressing to the lower layers (Tier-2 uplinks will be allocated from Tier-1 address blocks and so forth).

Server subnets on Tier-3 devices must be announced into BGP without using route summarization on Tier-2 and Tier-1 devices. Summarizing subnets in a Clos topology results in route black-holing under a single link failure (e.g., between Tier-2 and Tier-3 devices) and hence must be avoided. The use of peer links within the same tier to resolve the black-holing problem by providing "bypass paths" is undesirable due to O(N^2) complexity of the peering mesh and waste of ports on the devices. An alternative to the full-mesh of peer-links

would be using a simpler bypass topology, e.g., a "ring" as described in [FB4POST], but such a topology adds extra hops and has very limited bisectional bandwidth. Additionally requiring special tweaks to make BGP routing work - such as possibly splitting every device into an ASN on its own. Later in this document, Section 8.2 introduces a less intrusive method for performing a limited form of route summarization in Clos networks and discusses its associated trade-offs.

5.2.4. External Connectivity

A dedicated cluster (or clusters) in the Clos topology could be used for the purpose of connecting to the Wide Area Network (WAN) edge devices, or WAN Routers. Tier-3 devices in such cluster would be replaced with WAN routers, and EBGP peering would be used again, though WAN routers are likely to belong to a public ASN if Internet connectivity is required in the design. The Tier-2 devices in such a dedicated cluster will be referred to as "Border Routers" in this document. These devices have to perform a few special functions:

- Hide network topology information when advertising paths to WAN routers, i.e., remove Private Use ASNs [RFC6996] from the AS_PATH attribute. This is typically done to avoid ASN number collisions between different data centers and also to provide a uniform AS_PATH length to the WAN for purposes of WAN ECMP to Anycast prefixes originated in the topology. An implementation specific BGP feature typically called "Remove Private AS" is commonly used to accomplish this. Depending on implementation, the feature should strip a contiguous sequence of Private Use ASNs found in an AS_PATH attribute prior to advertising the path to a neighbor. This assumes that all ASNs used for intra data center numbering are from the Private Use ranges. The process for stripping the Private Use ASNs is not currently standardized, see [I-D.mitchell-grow-remove-private-as]. However most implementations at least follow the logic described in this vendor's document [VENDOR-REMOVE-PRIVATE-AS], which is enough for the design specified.

- Originate a default route to the data center devices. This is the only place where a default route can be originated, as route summarization is risky for the unmodified Clos topology. Alternatively, Border Routers may simply relay the default route learned from WAN routers. Advertising the default route from Border Routers requires that all Border Routers be fully connected to the WAN Routers upstream, to provide resistance to a single-link failure causing the black-holing of traffic. To prevent black-holing in the situation when all of the EBGP sessions to the WAN routers fail simultaneously on a given device, it is more

desirable to readvertise the default route rather than originating the default route via complicated conditional route origination schemes provided by some implementations [CONDITIONALROUTE].

5.2.5. Route Summarization at the Edge

It is often desirable to summarize network reachability information prior to advertising it to the WAN network due to high amount of IP prefixes originated from within the data center in a fully routed network design. For example, a network with 2000 Tier-3 devices will have at least 2000 servers subnets advertised into BGP, along with the infrastructure prefixes. However, as discussed before in Section 5.2.3, the proposed network design does not allow for route summarization due to the lack of peer links inside every tier.

However, it is possible to lift this restriction for the Border Routers, by devising a different connectivity model for these devices. There are two options possible:

- Interconnect the Border Routers using a full-mesh of physical links or using any other "peer-mesh" topology, such as ring or hub-and-spoke. Configure BGP accordingly on all Border Leafs to exchange network reachability information, e.g., by adding a mesh of IBGP sessions. The interconnecting peer links need to be appropriately sized for traffic that will be present in the case of a device or link failure in the mesh connecting the Border Routers.

- Tier-1 devices may have additional physical links provisioned toward the Border Routers (which are Tier-2 devices from the perspective of Tier-1). Specifically, if protection from a single link or node failure is desired, each Tier-1 devices would have to connect to at least two Border Routers. This puts additional requirements on the port count for Tier-1 devices and Border Routers, potentially making it a non-uniform, larger port count, device compared with the other devices in the Clos. This also reduces the number of ports available to "regular" Tier-2 switches and hence the number of clusters that could be interconnected via the Tier-1 layer.

If any of the above options are implemented, it is possible to perform route summarization at the Border Routers toward the WAN network core without risking a routing black-hole condition under a single link failure. Both of the options would result in non-uniform topology as additional links have to be provisioned on some network devices.

6. ECMP Considerations

This section covers the Equal Cost Multipath (ECMP) functionality for Clos topology and discusses a few special requirements.

6.1. Basic ECMP

ECMP is the fundamental load sharing mechanism used by a Clos topology. Effectively, every lower-tier device will use all of its directly attached upper-tier devices to load share traffic destined to the same IP prefix. The number of ECMP paths between any two Tier-3 devices in Clos topology is equal to the number of the devices in the middle stage (Tier-1). For example, Figure 5 illustrates a topology where Tier-3 device A has four paths to reach servers X and Y, via Tier-2 devices B and C and then Tier-1 devices 1, 2, 3, and 4 respectively.

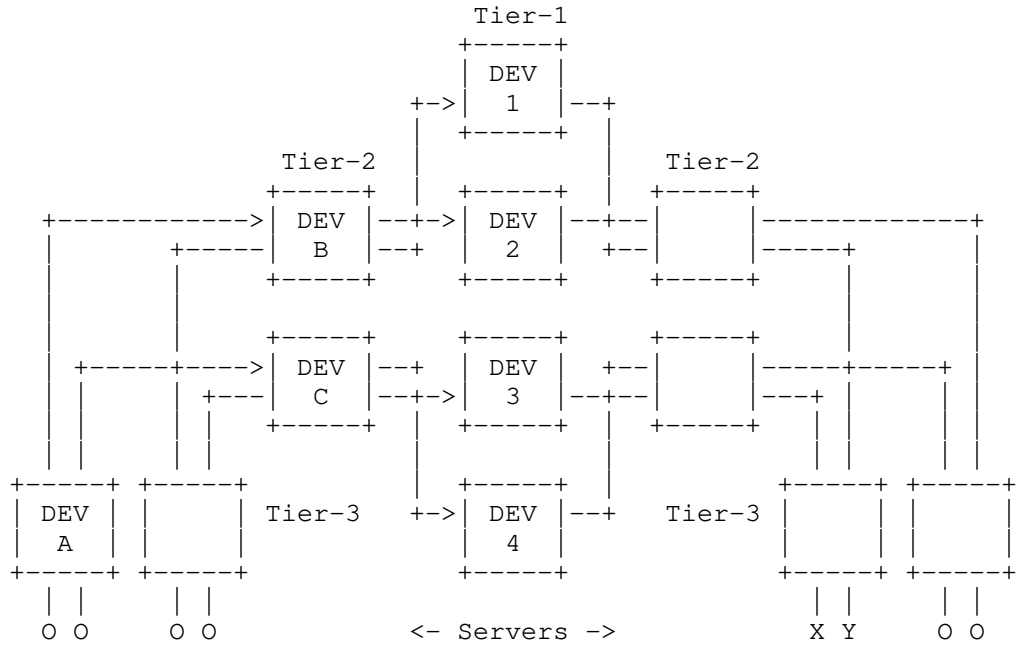

Figure 5: ECMP fan-out tree from A to X and Y

The ECMP requirement implies that the BGP implementation must support multipath fan-out for up to the maximum number of devices directly attached at any point in the topology in the upstream or downstream direction. Normally, this number does not exceed half of the ports found on a device in the topology. For example, an ECMP fan-out of 32 would be required when building a Clos network using 64-port

devices. The Border Routers may need to have wider fan-out to be able to connect to a multitude of Tier-1 devices if route summarization at Border Router level is implemented as described in Section 5.2.5. If a device's hardware does not support wider ECMP, logical link-grouping (link-aggregation at layer 2) could be used to provide "hierarchical" ECMP (Layer 3 ECMP coupled with Layer 2 ECMP) to compensate for fan-out limitations. However, this approach increases the risk of flow polarization, as less entropy will be available at the second stage of ECMP.

Most BGP implementations declare paths to be equal from an ECMP perspective if they match up to and including step (e) in Section 9.1.2.2 of [RFC4271]. In the proposed network design there is no underlying IGP, so all IGP costs are assumed to be zero or otherwise the same value across all paths and policies may be applied as necessary to equalize BGP attributes that vary in vendor defaults, such as MED and origin code. For historical reasons it is also useful to not use 0 as the equalized MED value; this and some other useful BGP information is available in [RFC4277] . Routing loops are unlikely due to the BGP best-path selection process which prefers shorter AS_PATH length, and longer paths through the Tier-1 devices which don't allow their own ASN in the path and have the same ASN are also not possible.

6.2. BGP ECMP over Multiple ASNs

For application load balancing purposes it is desirable to have the same prefix advertised from multiple Tier-3 devices. From the perspective of other devices, such a prefix would have BGP paths with different AS_PATH attribute values, while having the same AS_PATH attribute lengths. Therefore, BGP implementations must support load sharing over the above-mentioned paths. This feature is sometimes known as "multipath relax" or "multipath multiple-as" and effectively allows for ECMP to be done across different neighboring ASNs if all other attributes are equal as already described in the previous section.

6.3. Weighted ECMP

It may be desirable for the network devices to implement "weighted" ECMP, to be able to send more traffic over some paths in ECMP fan-out. This could be helpful to compensate for failures in the network and send more traffic over paths that have more capacity. The prefixes that require weighted ECMP would have to be injected using remote BGP speaker (central agent) over a multihop session as described further in Section 8.1. If support in implementations is available, weight-distribution for multiple BGP paths could be

signaled using the technique described in [I-D.ietf-idr-link-bandwidth].

6.4. Consistent Hashing

It is often desirable to have the hashing function used for ECMP to be consistent (see [CONS-HASH]), to minimize the impact on flow to next-hop affinity changes when a next-hop is added or removed to an ECMP group. This could be used if the network device is used as a load balancer, mapping flows toward multiple destinations - in this case, losing or adding a destination will not have a detrimental effect on currently established flows. One particular recommendation on implementing consistent hashing is provided in [RFC2992], though other implementations are possible. This functionality could be naturally combined with weighted ECMP, with the impact of the next-hop changes being proportional to the weight of the given next-hop. The downside of consistent hashing is increased load on hardware resource utilization, as typically more resources (e.g., TCAM space) are required to implement a consistent-hashing function.

7. Routing Convergence Properties

This section reviews routing convergence properties in the proposed design. A case is made that sub-second convergence is achievable if the implementation supports fast EBGP peering session deactivation and timely RIB and FIB update upon failure of the associated link.

7.1. Fault Detection Timing

BGP typically relies on an IGP to route around link/node failures inside an AS, and implements either a polling based or an event-driven mechanism to obtain updates on IGP state changes. The proposed routing design does not use an IGP, so the remaining mechanisms that could be used for fault detection are BGP keep-alive time-out (or any other type of keep-alive mechanism) and link-failure triggers.

Relying solely on BGP keep-alive packets may result in high convergence delays, on the order of multiple seconds (on many BGP implementations the minimum configurable BGP hold timer value is three seconds). However, many BGP implementations can shut down local EBGP peering sessions in response to the "link down" event for the outgoing interface used for BGP peering. This feature is sometimes called "fast fallover". Since links in modern data centers are predominantly point-to-point fiber connections, a physical interface failure is often detected in milliseconds and subsequently triggers a BGP re-convergence.

Ethernet links may support failure signaling or detection standards such as Connectivity Fault Management (CFM) as described in [IEEE8021Q], which may make failure detection more robust. Alternatively, some platforms may support Bidirectional Forwarding Detection (BFD) [RFC5880] to allow for sub-second failure detection and fault signaling to the BGP process. However, the use of either of these presents additional requirements to vendor software and possibly hardware, and may contradict REQ1. Until recently with [RFC7130], BFD also did not allow detection of a single member link failure on a LAG, which would have limited its usefulness in some designs.

7.2. Event Propagation Timing

In the proposed design the impact of the BGP Minimum Route Advertisement Interval (MRAI) timer (See section 9.2.1.1 of [RFC4271]) should be considered. Per the standard it is required for BGP implementations to space out consecutive BGP UPDATE messages by at least MRAI seconds, which is often a configurable value. The initial BGP UPDATE messages after an event carrying withdrawn routes are commonly not affected by this timer. The MRAI timer may present significant convergence delays when a BGP speaker "waits" for the new path to be learned from its peers and has no local backup path information.

In a Clos topology each EBGP speaker typically has either one path (Tier-2 devices don't accept paths from other Tier-2 in the same cluster due to same ASN) or N paths for the same prefix, where N is a significantly large number, e.g., N=32 (the ECMP fan-out to the next Tier). Therefore, if a link fails to another device from which a path is received there is either no backup path at all (e.g., from perspective of a Tier-2 switch losing the link to a Tier-3 device), or the backup is readily available in BGP Loc-RIB (e.g., from the perspective of a Tier-2 device losing the link to a Tier-1 switch). In the former case, the BGP withdrawal announcement will propagate without delay and trigger re-convergence on affected devices. In the latter case, the best-path will be re-evaluated and the local ECMP group corresponding to the new next-hop set changed. If the BGP path was the best-path selected previously, an "implicit withdraw" will be sent via a BGP UPDATE message as described as Option b in Section 3.1 of [RFC4271] due to the BGP AS_PATH attribute changing.

7.3. Impact of Clos Topology Fan-outs

Clos topology has large fan-outs, which may impact the "Up->Down" convergence in some cases, as described in this section. In a situation when a link between Tier-3 and Tier-2 device fails, the Tier-2 device will send BGP UPDATE messages to all upstream Tier-1

devices, withdrawing the affected prefixes. The Tier-1 devices, in turn, will relay these messages to all downstream Tier-2 devices (except for the originator). Tier-2 devices other than the one originating the UPDATE should then wait for ALL upstream Tier-1 devices to send an UPDATE message before removing the affected prefixes and sending corresponding UPDATE downstream to connected Tier-3 devices. If the original Tier-2 device or the relaying Tier-1 devices introduce some delay into their UPDATE message announcements, the result could be UPDATE message "dispersion", that could be as long as multiple seconds. In order to avoid such a behavior, BGP implementations must support "update groups". The "update group" is defined as a collection of neighbors sharing the same outbound policy - the local speaker will send BGP updates to the members of the group synchronously.

The impact of such "dispersion" grows with the size of topology fan-out and could also grow under network convergence churn. Some operators may be tempted to introduce "route flap dampening" type features that vendors include to reduce the control plane impact of rapidly flapping prefixes. However, due to issues described with false positives in these implementations especially under such "dispersion" events, it is not recommended to enable this feature in this design. More background and issues with "route flap dampening" and possible implementation changes that could affect this are well described in [RFC7196].

7.4. Failure Impact Scope

A network is declared to converge in response to a failure once all devices within the failure impact scope are notified of the event and have re-calculated their RIBs and consequently updated their FIBs. Larger failure impact scope typically means slower convergence since more devices have to be notified, and results in a less stable network. In this section we describe BGP's advantages over link-state routing protocols in reducing failure impact scope for a Clos topology.

BGP behaves like a distance-vector protocol in the sense that only the best path from the point of view of the local router is sent to neighbors. As such, some failures are masked if the local node can immediately find a backup path and does not have to send any updates further. Notice that in the worst case, all devices in a data center topology have to either withdraw a prefix completely or update the ECMP groups in their FIBs. However, many failures will not result in such a wide impact. There are two main failure types where impact scope is reduced:

- Failure of a link between Tier-2 and Tier-1 devices: In this case, a Tier-2 device will update the affected ECMP groups, removing the failed link. There is no need to send new information to downstream Tier-3 devices, unless the path was selected as best by the BGP process, in which case only an "implicit withdraw" needs to be sent, which should not affect forwarding. The affected Tier-1 device will lose the only path available to reach a particular cluster and will have to withdraw the associated prefixes. Such prefix withdrawal process will only affect Tier-2 devices directly connected to the affected Tier-1 device. The Tier-2 devices receiving the BGP UPDATE messages withdrawing prefixes will simply have to update their ECMP groups. The Tier-3 devices are not involved in the re-convergence process.

- Failure of a Tier-1 device: In this case, all Tier-2 devices directly attached to the failed node will have to update their ECMP groups for all IP prefixes from a non-local cluster. The Tier-3 devices are once again not involved in the re-convergence process, but may receive "implicit withdraws" as described above.

Even in the case of such failures where multiple IP prefixes will have to be reprogrammed in the FIB, it is worth noting that all of these prefixes share a single ECMP group on Tier-2 device. Therefore, in the case of implementations with a hierarchical FIB, only a single change has to be made to the FIB. Hierarchical FIB here means FIB structure where the next-hop forwarding information is stored separately from the prefix lookup table, and the latter only stores pointers to the respective forwarding information. See [I-D.ietf-rtgwg-bgp-pic] for discussion of FIB hierarchies and fast convergence.

Even though BGP offers reduced failure scope for some cases, further reduction of the fault domain using summarization is not always possible with the proposed design, since using this technique may create routing black-holes as mentioned previously. Therefore, the worst control plane failure impact scope is the network as a whole, for instance in the case of a link failure between Tier-2 and Tier-3 devices. The amount of impacted prefixes in this case would be much less than in the case of a failure in the upper layers of a Clos network topology. The property of having such large failure scope is not a result of choosing EBGP in the design but rather a result of using the Clos topology.

7.5. Routing Micro-Loops

When a downstream device, e.g., Tier-2 device, loses all paths for a prefix, it normally has the default route pointing toward the upstream device, in this case the Tier-1 device. As a result, it is

possible to get in the situation where a Tier-2 switch loses a prefix, but a Tier-1 switch still has the path pointing to the Tier-2 device, which results in a transient micro-loop, since the Tier-1 switch will keep passing packets to the affected prefix back to the Tier-2 device, and the Tier-2 will bounce them back again using the default route. This micro-loop will last for the duration of time it takes the upstream device to fully update its forwarding tables.

To minimize impact of such micro-loops, Tier-2 and Tier-1 switches can be configured with static "discard" or "null" routes that will be more specific than the default route for prefixes missing during network convergence. For Tier-2 switches, the discard route should be a summary route, covering all server subnets of the underlying Tier-3 devices. For Tier-1 devices, the discard route should be a summary covering the server IP address subnets allocated for the whole data center. Those discard routes will only take precedence for the duration of network convergence, until the device learns a more specific prefix via a new path.

8. Additional Options for Design

8.1. Third-party Route Injection

BGP allows for a "third-party", i.e., directly attached, BGP speaker to inject routes anywhere in the network topology, meeting REQ5. This can be achieved by peering via a multihop BGP session with some or even all devices in the topology. Furthermore, BGP diverse path distribution [RFC6774] could be used to inject multiple BGP next hops for the same prefix to facilitate load balancing, or using the BGP ADD-PATH capability [I-D.ietf-idr-add-paths] if supported by the implementation. Unfortunately, in many implementations ADD-PATH has been found to only support IBGP properly due to the use cases it was originally optimized for, which limits the "third-party" peering to IBGP only.

To implement route injection in the proposed design, a third-party BGP speaker may peer with Tier-3 and Tier-1 switches, injecting the same prefix, but using a special set of BGP next-hops for Tier-1 devices. Those next-hops are assumed to resolve recursively via BGP, and could be, for example, IP addresses on Tier-3 devices. The resulting forwarding table programming could provide desired traffic proportion distribution among different clusters.

8.2. Route Summarization within Clos Topology

As mentioned previously, route summarization is not possible within the proposed Clos topology since it makes the network susceptible to route black-holing under single link failures. The main problem is

the limited number of redundant paths between network elements, e.g., there is only a single path between any pair of Tier-1 and Tier-3 devices. However, some operators may find route aggregation desirable to improve control plane stability.

If any technique to summarize within the topology is planned, modeling of the routing behavior and potential for black-holing should be done not only for single or multiple link failures, but also fiber pathway failures or optical domain failures when the topology extends beyond a physical location. Simple modeling can be done by checking the reachability on devices doing summarization under the condition of a link or pathway failure between a set of devices in every tier as well as to the WAN routers when external connectivity is present.

Route summarization would be possible with a small modification to the network topology, though the trade-off would be reduction of the total size of the network as well as network congestion under specific failures. This approach is very similar to the technique described above, which allows Border Routers to summarize the entire data center address space.

8.2.1. Collapsing Tier-1 Devices Layer

In order to add more paths between Tier-1 and Tier-3 devices, group Tier-2 devices into pairs, and then connect the pairs to the same group of Tier-1 devices. This is logically equivalent to "collapsing" Tier-1 devices into a group of half the size, merging the links on the "collapsed" devices. The result is illustrated in Figure 6. For example, in this topology DEV C and DEV D connect to the same set of Tier-1 devices (DEV 1 and DEV 2), whereas before they were connecting to different groups of Tier-1 devices.

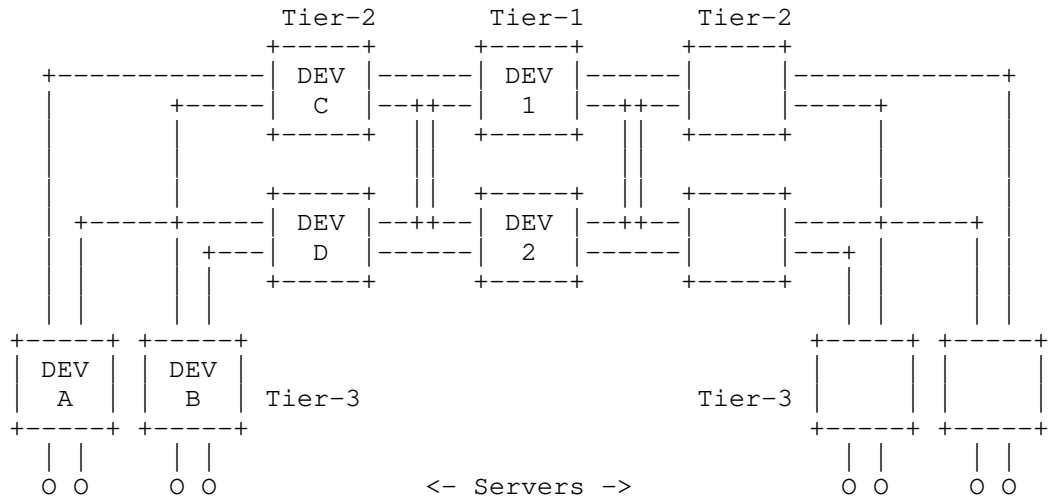

Figure 6: 5-Stage Clos topology

Having this design in place, Tier-2 devices may be configured to advertise only a default route down to Tier-3 devices. If a link between Tier-2 and Tier-3 fails, the traffic will be re-routed via the second available path known to a Tier-2 switch. It is still not possible to advertise a summary route covering prefixes for a single cluster from Tier-2 devices since each of them has only a single path down to this prefix. It would require dual-homed servers to accomplish that. Also note that this design is only resilient to single link failures. It is possible for a double link failure to isolate a Tier-2 device from all paths toward a specific Tier-3 device, thus causing a routing black-hole.

A result of the proposed topology modification would be a reduction of Tier-1 devices port capacity. This limits the maximum number of attached Tier-2 devices and therefore will limit the maximum DC network size. A larger network would require different Tier-1 devices that have higher port density to implement this change.

Another problem is traffic re-balancing under link failures. Since there are two paths from Tier-1 to Tier-3, a failure of the link between Tier-1 and Tier-2 switch would result in all traffic that was taking the failed link to switch to the remaining path. This will result in doubling the link utilization on the remaining link.

8.2.2. Simple Virtual Aggregation

A completely different approach to route summarization is possible, provided that the main goal is to reduce the FIB size, while allowing the control plane to disseminate full routing information. Firstly, it could be easily noted that in many cases multiple prefixes, some of which are less specific, share the same set of the next-hops (same ECMP group). For example, looking from the perspective of a Tier-3 devices, all routes learned from upstream Tier-2's, including the default route, will share the same set of BGP next-hops, provided that there are no failures in the network. This makes it possible to use the technique similar to described in [RFC6769] and only install the least specific route in the FIB, ignoring more specific routes if they share the same next-hop set. For example, under normal network conditions, only the default route needs to be programmed into the FIB.

Furthermore, if the Tier-2 devices are configured with summary prefixes covering all of their attached Tier-3 device's prefixes, the same logic could be applied in Tier-1 devices as well, and, by induction to Tier-2/Tier-3 switches in different clusters. These summary routes should still allow for more specific prefixes to leak to Tier-1 devices, to enable detection of mismatches in the next-hop sets if a particular link fails, changing the next-hop set for a specific prefix.

Re-stating once again, this technique does not reduce the amount of control plane state (i.e., BGP UPDATEs/BGP LocRIB size), but only allows for more efficient FIB utilization, by detecting more specific prefixes that share their next-hop set with a subsuming less specific prefix.

8.3. ICMP Unreachable Message Masquerading

This section discusses some operational aspects of not advertising point-to-point link subnets into BGP, as previously identified as an option in Section 5.2.3. The operational impact of this decision could be seen when using the well-known "traceroute" tool. Specifically, IP addresses displayed by the tool will be the link's point-to-point addresses, and hence will be unreachable for management connectivity. This makes some troubleshooting more complicated.

One way to overcome this limitation is by using the DNS subsystem to create the "reverse" entries for these point-to-point IP addresses pointing to the same name as the loopback address. The connectivity then can be made by resolving this name to the "primary" IP address of the devices, e.g., its Loopback interface, which is always

advertised into BGP. However, this creates a dependency on the DNS subsystem, which may be unavailable during an outage.

Another option is to make the network device perform IP address masquerading, that is rewriting the source IP addresses of the appropriate ICMP messages sent by the device with the "primary" IP address of the device. Specifically, the ICMP Destination Unreachable Message (type 3) codes 3 (port unreachable) and ICMP Time Exceeded (type 11) code 0, which are required for correct operation of the "traceroute" tool. With this modification, the "traceroute" probes sent to the devices will always be sent back with the "primary" IP address as the source, allowing the operator to discover the "reachable" IP address of the box. This has the downside of hiding the address of the "entry point" into the device. If the devices support [RFC5837], this may allow the best of both worlds by providing the information about the incoming interface even if the return address is the "primary" IP address.

9. Security Considerations

The design does not introduce any additional security concerns. General BGP security considerations are discussed in [RFC4271] and [RFC4272]. Since a DC is a single operator domain, this document assumes that edge filtering is in place to prevent attacks against the BGP sessions themselves from outside the perimeter of the DC. This may be a more feasible option for most deployments than having to deal with key management for TCP-MD5 as described in [RFC2385] or dealing with the lack of implementations available at the time of this document of [RFC5925]. The Generalized TTL Security Mechanism [RFC5082] could also be used to further reduce the risk of BGP session spoofing.

10. IANA Considerations

This document includes no request to IANA.

11. Acknowledgements

This publication summarizes work of many people who participated in developing, testing and deploying the proposed network design, some of whom were George Chen, Parantap Lahiri, Dave Maltz, Edet Nkposong, Robert Toomey, and Lihua Yuan. Authors would also like to thank Linda Dunbar, Anoop Ghanwani, Susan Hares, Danny McPherson, Robert Raszuk and Russ White for reviewing this document and providing valuable feedback and Mary Mitchell for initial grammar and style suggestions.

12. References

12.1. Normative References

[RFC4271] Rekhter, Y., Ed., Li, T., Ed., and S. Hares, Ed., "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, DOI 10.17487/RFC4271, January 2006, <http://www.rfc-editor.org/info/rfc4271>.

[RFC6996] Mitchell, J., "Autonomous System (AS) Reservation for Private Use", BCP 6, RFC 6996, DOI 10.17487/RFC6996, July 2013, <http://www.rfc-editor.org/info/rfc6996>.

12.2. Informative References

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

[RFC2385] Heffernan, A., "Protection of BGP Sessions via the TCP MD5 Signature Option", RFC 2385, DOI 10.17487/RFC2385, August 1998, <http://www.rfc-editor.org/info/rfc2385>.

[RFC2992] Hopps, C., "Analysis of an Equal-Cost Multi-Path Algorithm", RFC 2992, DOI 10.17487/RFC2992, November 2000, <http://www.rfc-editor.org/info/rfc2992>.

[RFC4272] Murphy, S., "BGP Security Vulnerabilities Analysis", RFC 4272, DOI 10.17487/RFC4272, January 2006, <http://www.rfc-editor.org/info/rfc4272>.

[RFC4277] McPherson, D. and K. Patel, "Experience with the BGP-4 Protocol", RFC 4277, DOI 10.17487/RFC4277, January 2006, <http://www.rfc-editor.org/info/rfc4277>.

[RFC4786] Abley, J. and K. Lindqvist, "Operation of Anycast Services", BCP 126, RFC 4786, DOI 10.17487/RFC4786, December 2006, <http://www.rfc-editor.org/info/rfc4786>.

[RFC5082] Gill, V., Heasley, J., Meyer, D., Savola, P., Ed., and C. Pignataro, "The Generalized TTL Security Mechanism (GTSM)", RFC 5082, DOI 10.17487/RFC5082, October 2007, <http://www.rfc-editor.org/info/rfc5082>.

[RFC5837] Atlas, A., Ed., Bonica, R., Ed., Pignataro, C., Ed., Shen, N., and JR. Rivers, "Extending ICMP for Interface and Next-Hop Identification", RFC 5837, DOI 10.17487/RFC5837, April 2010, <http://www.rfc-editor.org/info/rfc5837>.

[RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD)", RFC 5880, DOI 10.17487/RFC5880, June 2010, <http://www.rfc-editor.org/info/rfc5880>.

[RFC5925] Touch, J., Mankin, A., and R. Bonica, "The TCP Authentication Option", RFC 5925, DOI 10.17487/RFC5925, June 2010, <http://www.rfc-editor.org/info/rfc5925>.

[RFC6325] Perlman, R., Eastlake 3rd, D., Dutt, D., Gai, S., and A. Ghanwani, "Routing Bridges (RBridges): Base Protocol Specification", RFC 6325, DOI 10.17487/RFC6325, July 2011, <http://www.rfc-editor.org/info/rfc6325>.

[RFC6769] Raszuk, R., Heitz, J., Lo, A., Zhang, L., and X. Xu, "Simple Virtual Aggregation (S-VA)", RFC 6769, DOI 10.17487/RFC6769, October 2012, <http://www.rfc-editor.org/info/rfc6769>.

[RFC6774] Raszuk, R., Ed., Fernando, R., Patel, K., McPherson, D., and K. Kumaki, "Distribution of Diverse BGP Paths", RFC 6774, DOI 10.17487/RFC6774, November 2012, <http://www.rfc-editor.org/info/rfc6774>.

[RFC6793] Vohra, Q. and E. Chen, "BGP Support for Four-Octet Autonomous System (AS) Number Space", RFC 6793, DOI 10.17487/RFC6793, December 2012, <http://www.rfc-editor.org/info/rfc6793>.

[RFC7067] Dunbar, L., Eastlake 3rd, D., Perlman, R., and I. Gashinsky, "Directory Assistance Problem and High-Level Design Proposal", RFC 7067, DOI 10.17487/RFC7067, November 2013, <http://www.rfc-editor.org/info/rfc7067>.

[RFC7130] Bhatia, M., Ed., Chen, M., Ed., Boutros, S., Ed., Binderberger, M., Ed., and J. Haas, Ed., "Bidirectional Forwarding Detection (BFD) on Link Aggregation Group (LAG) Interfaces", RFC 7130, DOI 10.17487/RFC7130, February 2014, <http://www.rfc-editor.org/info/rfc7130>.

[RFC7196] Pelsser, C., Bush, R., Patel, K., Mohapatra, P., and O. Maennel, "Making Route Flap Damping Usable", RFC 7196, DOI 10.17487/RFC7196, May 2014, <http://www.rfc-editor.org/info/rfc7196>.

[I-D.ietf-idr-add-paths] Walton, D., Retana, A., Chen, E., and J. Scudder, "Advertisement of Multiple Paths in BGP", draft-ietf-idr-add-paths-15 (work in progress), May 2016.

[I-D.ietf-idr-link-bandwidth]
Mohapatra, P. and R. Fernando, "BGP Link Bandwidth Extended Community", draft-ietf-idr-link-bandwidth-06 (work in progress), January 2013.

[I-D.ietf-rtgwg-bgp-pic]
Bashandy, A., Filsfils, C., and P. Mohapatra, "Abstract", draft-ietf-rtgwg-bgp-pic-00 (work in progress), December 2015.

[I-D.mitchell-grow-remove-private-as]
Mitchell, J., Rao, D., and R. Raszuk, "Private Autonomous System (AS) Removal Requirements", draft-mitchell-grow-remove-private-as-04 (work in progress), April 2015.

[CLOS1953]
Clos, C., "A Study of Non-Blocking Switching Networks: Bell System Technical Journal Vol. 32(2)", March 1953.

[HADOOP] Apache, , "Apache HaDoop", April 2016, <https://hadoop.apache.org/>.

[GREENBERG2009]
Greenberg, A., Hamilton, J., and D. Maltz, "The Cost of a Cloud: Research Problems in Data Center Networks", January 2009.

[IEEE8021D-1990]
IEEE 802.1D, , "IEEE Standard for Local and Metropolitan Area Networks--Media access control (MAC) Bridges", May 1990.

[IEEE8021D-2004]
IEEE 802.1D, , "IEEE Standard for Local and Metropolitan Area Networks--Media access control (MAC) Bridges", February 2004.

[IEEE8021Q]
IEEE 802.1Q, , "IEEE Standard for Local and metropolitan area networks--Bridges and Bridged Networks", December 2014.

[INTERCON]
Dally, W. and B. Towles, "Principles and Practices of Interconnection Networks", ISBN 978-0122007514, January 2004.

[ALFARES2008]
Al-Fares, M., Loukissas, A., and A. Vahdat, "A Scalable, Commodity Data Center Network Architecture", August 2008.

[IANA.AS] IANA, , "Autonomous System (AS) Numbers", June 2016, <http://www.iana.org/assignments/as-numbers/>.

[IEEE8023AD]
IEEE 802.3ad, , "IEEE Standard for Link aggregation for parallel links", October 2000.

[ALLOWASIN]
Cisco Systems, , "Allowas-in Feature in BGP Configuration Example", June 2016, <http://www.cisco.com/c/en/us/support/docs/ip/border-gateway-protocol-bgp/112236-allowas-in-bgp-config-example.html>.

[VENDOR-REMOVE-PRIVATE-AS]
Cisco Systems, , "Removing Private Autonomous System Numbers in BGP", August 2005, <http://www.cisco.com/en/US/tech/tk365/technologies_tech_note09186a0080093f27.shtml>.

[CONDITIONALROUTE]
Cisco Systems, , "Configuring and Verifying the BGP Conditional Advertisement Feature", August 2005, <http://www.cisco.com/c/en/us/support/docs/ip/border-gateway-protocol-bgp/16137-cond-adv.html>.

[FB4POST] Farrington, N. and A. Andreyev, "Facebook's Data Center Network Architecture", May 2013, <http://nathanfarrington.com/papers/facebook-oic13.pdf>.

[JAKMA2008]
Jakma, P., "BGP Path Hunting", 2008, <https://blogs.oracle.com/paulj/entry/bgp_path_hunting>.

[CONS-HASH]
Wikipedia, , "Consistent Hashing", <http://en.wikipedia.org/wiki/Consistent_hashing>.

[L3DSR] Schaumann, J., "L3DSR - Overcoming Layer 2 Limitations of Direct Server Return Load Balancing", 2011, <https://www.nanog.org/meetings/nanog51/presentations/Monday/NANOG51.Talk45.nanog51-Schaumann.pdf>.

Authors' Addresses

Petr Lapukhov
Facebook
1 Hacker Way
Menlo Park, CA 94025
US

Email: petr@fb.com

Ariff Premji
Arista Networks
5453 Great America Parkway
Santa Clara, CA 95054
US

Email: ariff@arista.com
URI: http://arista.com/

Jon Mitchell (editor)

Email: jrmitche@puck.nether.net

rtgwg
Internet-Draft
Intended status: Standards Track
Expires: December 29, 2015

D. Lamparter
NetDEF

June 27, 2015

Destination/Source Routing
draft-lamparter-rtgwg-dst-src-routing-01

Abstract

This note specifies using packets' source addresses in route lookups as additional qualifier to be used in route lookup. This applies to IPv6 [RFC2460] in general with specific considerations for routing protocol left for separate documents.

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 http://datatracker.ietf.org/drafts/current/.

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

This Internet-Draft will expire on December 29, 2015.

Table of Contents

1. Introduction

Since connectivity providers generally secure their ingress along the lines of BCP 38 [RFC2827], small multihomed networks have a need to ensure their traffic leaves their network with a correct combination of source address and exit taken. This applies to networks of a particular pattern where the provider's default (dynamic) address provisioning methods are used and no fixed IP space is allocated, e.g. home networks, small business users and mobile ad-hoc setups.

While IPv4 networks would conventionally use NAT or policy routing to produce correct behaviour, this not desirable to carry over to IPv6. Instead, assigning addresses from multiple prefixes in parallel shifts the choice of uplink to the host. However, now for finding the proper exit the source address of packets must be taken into account.

For a general introduction and aspects of interfacing routers to hosts, refer to [I-D.sarikaya-6man-sadr-overview].

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. Principle of operation

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.

2.1. Lookup ordering and disambiguation

Adding further criteria to be looked up when forwarding packets on a hop-by-hop basis has the very fundamental requirement that all routers behave the same way in choosing the most specific route when there are multiple eligible routes.

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 :=    Adst <  Bdst
        || (Adst == Bdst && Asrc < Bsrc)
```

2.2. Ordering Rationale

The ordering described by this document (destination before source) could as well be reversed, which would lead to semantically different behavior.

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 lookup.

3. Applicability To Specific Situations

3.1. Recursive Route Lookups

TBD, multiple possible approaches:

variant 1: ignore dst-src routes, only use routes with src ::/0

variant 2: exact-match src prefixes from resolvee to resolvent (will not work for a lot of cases)

variant 3: longer-match src prefixes from resolvee to resolvent (nexthop src may be superset of looked-up route)

variant 4: create multiple instances of the route whose nexthop is resolved, with different source prefixes

(Variant 4:)

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,                                              via fe80::1
2001:db8:abcd::/48,                                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
```

3.2. 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.

3.3. 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.

4. Interoperability

Since a router implementing source/destination routing can have additional, more specific routes than one that doesn't implement source/destination routing, persistent loops can form between these systems. To prevent this from happening, a simple rule must be followed:

The set of qualifiers used to route a particular packet MUST be a subset of the qualifiers supported by the next hop.

This means in particular that a router using the source address as extra qualifier MUST NOT route packets based on a source/destination route to a system that doesn't support source/destination routes (and hence doesn't understand the route).

There are 3 possible approaches to avoid such a condition:

1. discard the packet (treat as destination unreachable)

2. calculate an alternate topology including only routers that support qualifier A

3. if the lookup returns the same nexthop without using qualifier A, use that result (i.e., the nexthop is known to correctly route the packet)

Above considerations require under all circumstances a knowledge of the next router's capabilities. For routing protocols based on hop-by-hop flooding (RIP [RFC2080], BGP [RFC4271]), knowing the peer's capabilities - or simply relying on systems to only flood what they understand - is sufficient. Protocols building a link-state database (OSPF [RFC5340], IS-IS [RFC5308]) have the additional opportunity to calculate alternate paths based on knowledge of the entire domain, but cannot rely on routers flooding only link state they support themselves.

5. IANA Considerations

This document makes no requests to IANA.

6. 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 source/destination 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 source/destination route, and in that sense could be considered similar to an access list that is managed by and scales with routing.

7. 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.

8. 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.

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

9. Change Log

Initial Version: April 2015: merged routing-extra-qualifiers draft, new ordering rationale section

Initial Version: October 2014

10. References

10.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2460] Deering, S.E. and R.M. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.

10.2. Informative References

[I-D.baker-ipv6-isis-dst-src-routing]
Baker, F., "IPv6 Source/Destination Routing using IS-IS", draft-baker-ipv6-isis-dst-src-routing-01 (work in progress), August 2013.

[I-D.sarikaya-6man-sadr-overview]

Sarikaya, B., "Overview of Source Address Dependent Routing", draft-sarikaya-6man-sadr-overview-01 (work in progress), September 2014.

[I-D.troan-homenet-sadr]
Troan, O. and L. Colitti, "IPv6 Multihoming with Source Address Dependent Routing (SADR)", draft-troan-homenet-sadr-01 (work in progress), September 2013.

[RFC2080] Malkin, G. and R. Minnear, "RIPng for IPv6", RFC 2080, January 1997.

[RFC2827] Ferguson, P. and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", BCP 38, RFC 2827, May 2000.

[RFC4271] Rekhter, Y., Li, T., and S. Hares, "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, January 2006.

[RFC5308] Hopps, C., "Routing IPv6 with IS-IS", RFC 5308, October 2008.

[RFC5340] Coltun, R., Ferguson, D., Moy, J., and A. Lindem, "OSPF for IPv6", RFC 5340, July 2008.

Author's Address

David Lamparter
NetDEF
Leipzig 04103
Germany

Email: david@opensourcerouting.org

rtgwg
Internet-Draft
Intended status: Standards Track
Expires: April 23, 2015

D. Lamparter
NetDEF

October 20, 2014

Considerations and Registry for extending IP route lookup draft-lamparter-rtgwg-routing-extra-qualifiers-00

Abstract

This document describes the behaviour of a routing system that takes additional specifications on routes--extra qualifiers--into account on a hop-by-hop basis, augmenting longest match behaviour.

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 http://datatracker.ietf.org/drafts/current/.

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

This Internet-Draft will expire on April 23, 2015.

Table of Contents

1. Introduction

IP Routing systems at the time of creation of this document are occasionally already capable of matching more than the packet's destination addresses to lookup routes, preexisting patterns include virtual routers (i.e. keying by routing instance), QoS-aware routing (keying by DSCP bits) and the relatively unspecific "policy routing."

Additional developments extend this field to the point where a lack of well-defined specification may lead to interoperability problems. The intent of this document is to construct a reference framework for extensions on the match aspect of IP routes.

Specifically, since IP Routing includes longest-match route selection, the ordering of all match qualifiers must be the same among all routers to prevent loops or connectivity loss.

While this document is written with IPv6 in mind, it applies to IP router architecture in general, including IPv4 routers.

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. Applicability

While the conceptually same longest-prefix routing is used not only for routing packets, but also recursive route/nexthop lookups, multicast reverse path forwarding and unicast reverse path filtering. However, while based on the same base principle, these applications may differ in their requirements. For example, multicast RPF cannot use source address discriminators since no source address is known at the time of lookup.

The intent of this specification is only to provide a basic framework; individual extensions to route match behaviour MUST clarify their respective applicability.

3. Match criteria (informational)

3.1. Virtual routers

While not documented to this extent, an implementation capable of partitioning a physical router into multiple virtual routers is an application that essentially has the virtual router identifier as first key in lookup operations. This may not be implemented as such, for example by keeping tables completely separate, however the end behaviour is the same; lookups are made local to the router instance.

3.2. Policy routing

Equally little specified as virtual routers, policy routing usually applies certain qualifiers (source address, traffic class, firewall markers) prior to destination address match.

3.3. Destination address longest match

The conventional destination IP address longest match is included at this point as it is, barring implementation specific extensions mentioned above, the first qualifier used to match packets against the route table.

3.4. Source address longest match

Currently under development, matching on the source address permits routers to choose the correct (in terms of [RFC2827]) exit in smaller multihomed networks. This is distinct from policy routing in that only few select (usually default) routes would be annotated with source prefixes.

Various aspects of this are described in:

[I-D.troan-homenet-sadr]

[I-D.boutier-homenet-source-specific-routing]

[I-D.sarikaya-6man-sadr-overview]

[I-D.baker-rtgwg-src-dst-routing-use-cases]

[I-D.baker-ipv6-isis-dst-src-routing]

[I-D.baker-ipv6-ospf-dst-src-routing]

[I-D.baker-rtgwg-src-dst-routing-use-cases]

3.5. Flowlabel routing

TBD, described in:

[I-D.baker-ipv6-isis-dst-flowlabel-routing]

[I-D.baker-ipv6-ospf-dst-flowlabel-routing]

3.6. QoS/DSCP traffic class based routing

TBD (deprecated, reference only)

4. Requirements to extending match behaviour

4.1. Match ordering

Adding further criteria to be looked up when forwarding packets on a hop-by-hop basis has the very fundamental requirement that all routers behave the same way in choosing the most specific route when there are multiple eligible routes.

This document disambiguates this situation by recording the order of specificness in a registry. This means that the comparison for "more specific", here indicated by A < B (to mean A is more specific than B), is redefined as concatenation for attributes a, b, c as:

```
A < B :=    Aa <  Ba
         || (Aa == Ba && Ab <  Bb)
         || (Aa == Ba && Ab == Bb && Ac < Bc )
```

This transfers to a sample situation (using source address, destination address and flowlabel as qualifiers):

Example route table

	destination	source	flowlabel
route A:	2001:db8::/32		
route B:	2001:db8:1234::/48	2001:db8:4567::/48	
route C:	2001:db8:1234::/48		abcde
route D:	2001:db8:1234:5678::/64	2001:db8:4567::/48	abcde
route E:	2001:db8:1234:5678::/64		

Showing the different results between "destination, source, flowlabel" ("DSF") and "destination, flowlabel, source" ("DFS") ordering:

Example match results

packet to be routed				result	
#	destination	source	flowlabel	"DSF"	"DFS"
1	2001:db8::1	2001:db8:4567::1	abcde	A	A
2	2001:db8:1234::1	2001:db8:4567::1	abcde	B	C
3	2001:db8:1234::1	2001:db8:4567::1	11111	B	B
4	2001:db8:1234::1	2001:db8:1111::1	abcde	C	C
5	2001:db8:1234::1	2001:db8:1111::1	11111	A	A
6	2001:db8:1234:5678::1	2001:db8:4567::1	abcde	D	D
7	2001:db8:1234:5678::1	2001:db8:4567::1	11111	E	E
8	2001:db8:1234:5678::1	2001:db8:1111::1	abcde	E	E

It should be noted that lookup may not result in usage of the most specific element even for the first attribute (destination in the example). As displayed in #5 above, the route used is the most specific one that satisfies all conditions. If a system cannot "back out" to less specific matches on earlier attributes, this MUST be worked around by installing synthetic routes for these cases.

4.2. Compatibility / Interoperability

Since a router implementing extra match qualifiers can have additional, more specific routes than one that doesn't implement these qualifiers, persistent loops can form between these systems. To prevent this from happening, a simple rule must be followed:

The set of qualifiers used to route a particular packet MUST be a subset of the qualifiers supported by the next hop.

This means in particular that a router using extra qualifier A MUST NOT route packets based on a route that checks this qualifier to a system that doesn't support qualifier A (and hence doesn't understand the route).

There are 3 possible approaches to avoid such a condition:

1. discard the packet (treat as destination unreachable)

2. calculate an alternate topology including only routers that support qualifier A

3. if the lookup returns the same nexthop without using qualifier A, use that result (i.e., the nexthop is known to correctly route the packet)

Above considerations require under all circumstances a knowledge of the next router's capabilities. For routing protocols based on hop-by-hop flooding (RIP [RFC2080], BGP [RFC4271]), knowing the peer's capabilities - or simply relying on systems to only flood what they understand - is sufficient. Protocols building a link-state database (OSPF [RFC5340], IS-IS [RFC5308]) have the additional opportunity to calculate alternate paths based on knowledge of the entire domain, but cannot rely on routers flooding only link state they support themselves.

5. IANA Considerations

This document requests creation of a new registry called the "Routing Qualifier Registry." The registry consists of an ordered list of items, no identifier value needs to be assigned. The only purpose of the registry is to document the order in which qualifiers are evaluated.

Registry items must specify the following information:

- Name of the qualifier

- Applicable protocols (IP version 4 and/or IP version 6)

- Specification reference (possibly distinct between IPv4 and IPv6)
- Insertion position, listing both the previous and next entry to avoid confusion

The allocation policy per [RFC5226] is "IETF Review." This is intended to help keep routing systems compatible with each other.

5.1. Initial list

The list is prepropagated with a single entry describing "classical" destination-based routing:

Name: Destination lookup

Applicable to IPv4 and IPv6

Specification references: [RFC4632] for IPv4, [RFC2460] for IPv6

6. Security Considerations

This document specifies only the ordering of lookups. Making no change to the existing situation, there are no security considerations for this document.

7. Privacy Considerations

As with security considerations, no privacy considerations apply to this document.

Introducing additional routing qualifiers has the potential to expose information that was not previously visible, in particular if such information would otherwise be scrubbed by a process like NAT. However, these considerations are left for documents actually introducing new routing qualifiers.

8. Acknowledgements

This document is largely the result of discussions with Fred Baker.

A lot of drafts exists in this general area, refer to the informative references section below.

9. Change Log

Initial Version: October 2014

10. References

10.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2460] Deering, S.E. and R.M. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.

[RFC4632] Fuller, V. and T. Li, "Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan", BCP 122, RFC 4632, August 2006.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

10.2. Informative References

[I-D.baker-ipv6-isis-dst-flowlabel-routing]
Baker, F., "Using IS-IS with Token-based Access Control", draft-baker-ipv6-isis-dst-flowlabel-routing-01 (work in progress), August 2013.

[I-D.baker-ipv6-isis-dst-src-routing]
Baker, F., "IPv6 Source/Destination Routing using IS-IS", draft-baker-ipv6-isis-dst-src-routing-01 (work in progress), August 2013.

[I-D.baker-ipv6-ospf-dst-flowlabel-routing]
Baker, F., "Using OSPFv3 with Token-based Access Control", draft-baker-ipv6-ospf-dst-flowlabel-routing-03 (work in progress), August 2013.

[I-D.baker-ipv6-ospf-dst-src-routing]
Baker, F., "IPv6 Source/Destination Routing using OSPFv3", draft-baker-ipv6-ospf-dst-src-routing-03 (work in progress), August 2013.

[I-D.baker-rtgwg-src-dst-routing-use-cases]
Baker, F., "Requirements and Use Cases for Source/ Destination Routing", draft-baker-rtgwg-src-dst-routing-use-cases-00 (work in progress), August 2013.

[I-D.boutier-homenet-source-specific-routing]
Boutier, M. and J. Chroboczek, "Source-specific Routing", draft-boutier-homenet-source-specific-routing-00 (work in progress), July 2013.

[I-D.sarikaya-6man-sadr-overview]
Sarikaya, B., "Overview of Source Address Dependent Routing", draft-sarikaya-6man-sadr-overview-01 (work in progress), September 2014.

[I-D.troan-homenet-sadr]
Troan, O. and L. Colitti, "IPv6 Multihoming with Source Address Dependent Routing (SADR)", draft-troan-homenet-sadr-01 (work in progress), September 2013.

[RFC2080] Malkin, G. and R. Minnear, "RIPng for IPv6", RFC 2080, January 1997.

[RFC2827] Ferguson, P. and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", BCP 38, RFC 2827, May 2000.

[RFC4271] Rekhter, Y., Li, T., and S. Hares, "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, January 2006.

[RFC5308] Hopps, C., "Routing IPv6 with IS-IS", RFC 5308, October 2008.

[RFC5340] Coltun, R., Ferguson, D., Moy, J., and A. Lindem, "OSPF for IPv6", RFC 5340, July 2008.

Author's Address

David Lamparter
NetDEF
Leipzig 04103
Germany

Email: david@opensourcerouting.org

Routing Area Working Group
Internet-Draft
Intended status: Informational
Expires: January 9, 2017

P. Sarkar, Ed.
Individual
S. Hegde
C. Bowers
Juniper Networks, Inc.
U. Chunduri, Ed.
Ericsson Inc.
J. Tantsura
Individual
B. Decraene
Orange
H. Gredler
Unaffiliated

July 8, 2016

LFA selection for Multi-Homed Prefixes
draft-psarkar-rtgwg-multihomed-prefix-lfa-04

Abstract

This document shares experience gained from implementing algorithms to determine Loop-Free Alternates for multi-homed prefixes. In particular, this document provides explicit inequalities that can be used to evaluate neighbors as a potential alternates for multi-homed prefixes. It also provides detailed criteria for evaluating potential alternates for external prefixes advertised by OSPF ASBRs.

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 [RFC2119].

Status of This Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

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

This Internet-Draft will expire on January 9, 2017.

Table of Contents

1. Introduction

The use of Loop-Free Alternates (LFA) for IP Fast Reroute is specified in [RFC5286]. Section 6.1 of [RFC5286] describes a method to determine loop-free alternates for a multi-homed prefixes (MHPs). This document describes a procedure using explicit inequalities that can be used by a computing router to evaluate a neighbor as a potential alternate for a multi-homed prefix. The results obtained are equivalent to those obtained using the method described in Section 6.1 of [RFC5286]. However, some may find this formulation useful.

Section 6.3 of [RFC5286] discusses complications associated with computing LFAs for multi-homed prefixes in OSPF. This document provides detailed criteria for evaluating potential alternates for external prefixes advertised by OSPF ASBRs, as well as explicit inequalities.

This document also provide clarifications, additional considerations to [RFC5286], to address a few coverage and operational observations. These observations are in the area of handling IS-IS attach (ATT) bit in Level-1 (L1) area, links provisioned with MAX_METRIC for traffic engineering (TE) purposes and in the area of Multi Topology (MT) IGP deployments. All these are elaborated in detail in Section 3.2 and Section 5.

1.1. Acronyms

AF - Address Family

ATT - IS-IS Attach Bit

ECMP - Equal Cost Multi Path

IGP - Interior Gateway Protocol

IS-IS - Intermediate System to Intermediate System

OSPF - Open Shortest Path First

MHP - Multi-homed Prefix

MT - Multi Topology

SPF - Shortest Path First PDU

2. LFA inequalities for MHPs

This document proposes the following set of LFA inequalities for selecting the most appropriate LFAs for multi-homed prefixes (MHPs). They can be derived from the inequalities in [RFC5286] combined with the observation that D_opt(N,P) = Min (D_opt(N,PO_i) + cost(PO_i,P)) over all PO_i

Link-Protection:
D_opt(N,PO_i)+ cost(PO_i,P) < D_opt(N,S) +
D_opt(S,PO_best) + cost(PO_best,P)

Link-Protection + Downstream-paths-only:
D_opt(N,PO_i)+ cost(PO_i,P) < D_opt(S,PO_best) + cost(PO_best,P)

Node-Protection:
D_opt(N,PO_i)+ cost(PO_i,P) < D_opt(N,E) +
D_opt(E,PO_best) + cost(PO_best,P)

Where,

S	- The computing router
N	- The alternate router being evaluated
E	- The primary next-hop on shortest path from S to prefix P.
PO_i	- The specific prefix-originating router being evaluated.
PO_best	- The prefix-originating router on the shortest path from the computing router S to prefix P.
Cost (X,P)	- Cost of reaching the prefix P from prefix originating node X.
D_opt(X,Y)	- Distance on the shortest path from node X to node Y.

Figure 1: LFA inequalities for MHPs

3. LFA selection for the multi-homed prefixes

To compute a valid LFA for a given multi-homed prefix P, through an alternate neighbor N a computing router S MUST follow one of the appropriate procedures below.

```
Link-Protection :
=================
1. If alternate neighbor N is also prefix-originator of P,
   1.a. Select N as a LFA for prefix P (irrespective of
        the metric advertised by N for the prefix P).
2. Else, evaluate the link-protecting LFA inequality for P with
   the N as the alternate neighbor.
   2.a. If LFA inequality condition is met,
        select N as a LFA for prefix P.
   2.b. Else, N is not a LFA for prefix P.

Link-Protection + Downstream-paths-only :
=========================================
1. Evaluate the link-protecting + downstream-only LFA inequality
   for P with the N as the alternate neighbor.
   1.a. If LFA inequality condition is met,
        select N as a LFA for prefix P.
   1.b. Else, N is not a LFA for prefix P.

Node-Protection :
=================
1. If alternate neighbor N is also prefix-originator of P,
   1.a. Select N as a LFA for prefix P (irrespective of
        the metric advertised by N for the prefix P).
2. Else, evaluate the apporpriate node-protecting LFA inequality
   for P with the N as the alternate neighbor.
   2.a. If LFA inequality condition is met,
        select N as a LFA for prefix P.
   2.b. Else, N is not a LFA for prefix P.
```

Figure 2: Rules for selecting LFA for MHPs

In case an alternate neighbor N is also one of the prefix-originators of prefix P, N MAY be selected as a valid LFA for P.

However if N is not a prefix-originator of P, the computing router SHOULD evaluate one of the corresponding LFA inequalities, as mentioned in Figure 1, once for each remote node that originated the prefix. In case the inequality is satisfied by the neighbor N router S MUST choose neighbor N, as one of the valid LFAs for the prefix P.

When computing a downstream-only LFA, in addition to being a prefix-originator of P, router N MUST also satisfy the downstream-only LFA inequality specified in Figure 1.

For more specific rules please refer to the later sections of this document.

3.1. Improved coverage with simplified approach to MHPs

LFA base specification [RFC5286] Section 6.1 recommends that a router compute the alternate next-hop for an IGP multi-homed prefix by considering alternate paths via all routers that have announced that prefix and the same has been elaborated with appropriate inequalities in the above section. However, [RFC5286] Section 6.1 also allows for the router to simplify the multi-homed prefix calculation by assuming that the MHP is solely attached to the router that was its pre-failure optimal point of attachment, at the expense of potentially lower coverage. If an implementation chooses to simplify the multi-homed prefix calculation by assuming that the MHP is solely attached to the router that was its pre-failure optimal point of attachment, the procedure described in this memo can potentially improve coverage for equal cost multi path (ECMP) MHPs without incurring extra computational cost.

While the approach as specified in [RFC5286] Section 6.1 last paragraph, is to simplify the MHP as solely attached to the router that was its pre-failure optimal point of attachment; though it is a scalable approach and simplifies computation, [RFC5286] notes this may result in little less coverage.

This memo improves the above approach to provide loop-free alternatives without any additional cost for equal cost multi path MHPs as described through the below example network. The approach specified here MAY also be applicable for handling default routes as explained in Section 3.2.

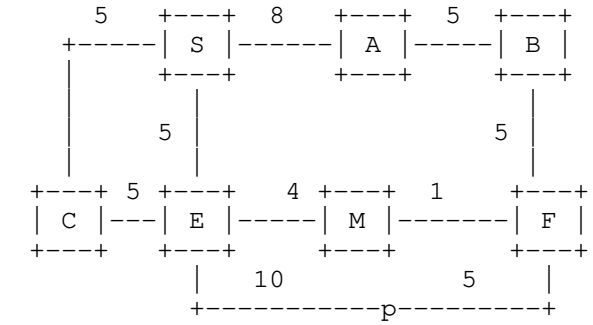

Figure 3: MHP with same ECMP Next-hop

In the above network a prefix p, is advertised from both Node E and Node F. With simplified approach taken as specified in [RFC5286] Section 6.1, prefix p will get only link protection LFA through the neighbor C while a node protection path is available through neighbor

A. In this scenario, E and F both are pre-failure optimal points of attachment and share the same primary next-hop. Hence, an implementation MAY compare the kind of protection A provides to F (link-and-node protection) with the kind of protection C provides to E (link protection) and inherit the better alternative to prefix p and here it is A.

However, in the below network prefix p has an ECMP through both node E and node F with cost 20. Though it has 2 pre-failure optimal points of attachment, the primary next-hop to each pre-failure optimal point of attachment is different. In this case, prefix p shall inherit corresponding LFA to each primary next-hop calculated for the router advertising the same respectively (node E's and node F's LFA).

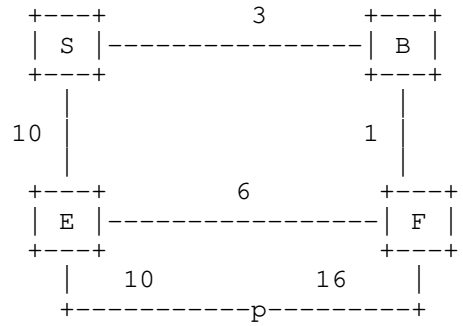

Figure 4: MHP with different ECMP Next-hops

In summary, if there are multiple pre-failure points of attachment for a MHP and primary next-hop of a MHP is same as that of the primary next-hop of the router that was pre-failure optimal point of attachment, an implementation MAY provide the better protection to MHP without incurring any additional computation cost.

3.2. IS-IS ATT Bit considerations

Per [RFC1195] a default route needs to be added in Level1 (L1) router to the closest reachable Level1/Level2 (L1/L2) router in the network advertising ATT (attach) bit in its LSP-0 fragment. All L1 routers in the area would do this during the decision process with the next-hop of the default route set to the adjacent router through which the closest L1/L2 router is reachable. The base LFA specification [RFC5286] does not specify any procedure for computing LFA for a default route in IS-IS L1 area. Potentially one MAY consider a default route is being advertised from the border L1/L2 router where ATT bit is set and can do LFA computation for the default route.

But, when multiple ECMP L1/L2 routers are reachable in an L1 area corresponding best LFAs SHOULD be given for each primary next-hop associated with default route. Considerations as specified in Section 3 and Section 3.1 are applicable for default routes, if the default route is considered as ECMP MHP.

4. LFA selection for the multi-homed external prefixes

Redistribution of external routes into IGP is required in case of two different networks getting merged into one or during protocol migrations. External routes could be distributed into an IGP domain via multiple nodes to avoid a single point of failure.

During LFA calculation, alternate LFA next-hops to reach the best ASBR could be used as LFA for the routes redistributed via that ASBR. When there is no LFA available to the best ASBR, it may be desirable to consider the other ASBRs (referred to as alternate ASBR hereafter) redistributing the external routes for LFA selection as defined in [RFC5286] and leverage the advantage of having multiple re-distributing nodes in the network.

4.1. IS-IS

LFA evaluation for multi-homed external prefixes in IS-IS is similar to the multi-homed internal prefixes. Inequalities described in sec 2 would also apply to multi-homed external prefixes as well.

4.2. OSPF

Loop free Alternates [RFC 5286] describes mechanisms to apply inequalities to find the loop free alternate neighbor. For the selection of alternate ASBR for LFA consideration, additional rules have to be applied in selecting the alternate ASBR due to the external route calculation rules imposed by [RFC 2328].

This document also defines the inequalities defined in RFC [5286] specifically for the alternate loop-free ASBR evaluation.

4.2.1. Rules to select alternate ASBR

The process to select an alternate ASBR is best explained using the rules below. The below process is applied when primary ASBR for the concerned prefix is chosen and there is an alternate ASBR originating same prefix.

1. If RFC1583Compatibility is disabled

   1a. if primary ASBR and alternate ASBR are intra area
       non-backbone path go to step 2.
   1b. If primary ASBR and alternate ASBR belong to
       intra-area backbone and/or inter-area path go
       to step 2.
   1c. for other paths, skip the alternate ASBR and
       consider next ASBR.

2. If cost type (type1/type2) advertised by alternate
   ASBR same as primary
   2a. If not same skip alternate ASBR and consider next ASBR.

3. If cost type is type1
   3a. If cost is same, program ECMP
   3b. else go to step 5.

4 If cost type is type 2
   4a. If cost is different, skip alternate ASBR and
       consider next ASBR
   4b. If type2 cost is same, compare type 1 cost.
   4c. If type1 cost is also same program ECMP.
   4d. If type 1 cost is different go to step 5.

5. If route type (type 5/type 7)
   5a. If route type is same, check route p-bit,
       forwarding address field for routes from both
       ASBRs
       match. If not skip alternate ASBR and consider
       next ASBR.
   5b. If route type is not same, skip ASBR
       and consider next ASBR.

6. Apply inequality on the alternate ASBR.

Figure 5: Rules for selecting alternate ASBR in OSPF

4.2.2. Multiple ASBRs belonging different area

When "RFC1583compatibility" is set to disabled, OSPF[RFC2328] defines certain rules of preference to choose the ASBRs. While selecting alternate ASBR for loop evaluation for LFA, these rules should be applied and ensured that the alternate neighbor does not loop the traffic back.

When there are multiple ASBRs belonging to different area advertising the same prefix, pruning rules as defined in RFC 2328 section 16.4.1

are applied. The alternate ASBRs pruned using above rules are not considered for LFA evaluation.

4.2.3. Type 1 and Type 2 costs

If there are multiple ASBRs not pruned via rules defined in 3.2.2, the cost type advertised by the ASBRs is compared. ASBRs advertising Type1 costs are preferred and the type2 costs are pruned. If two ASBRs advertise same type2 cost, the alternate ASBRs are considered along with their type1 cost for evaluation. If the two ASBRs with same type2 as well as type1 cost, ECMP FRR is programmed. If there are two ASBRs with different type2 cost, the higher cost ASBR is pruned. The inequalities for evaluating alternate ASBR for type 1 and type 2 costs are same, as the alternate ASBRs with different type2 costs are pruned and the evaluation is based on equal type 2 cost ASBRS.

4.2.4. RFC1583compatibility is set to enabled

When RFC1583Compatibility is set to enabled, multiple ASBRs belonging to different area advertising same prefix are chosen based on cost and hence are valid alternate ASBRs for the LFA evaluation.

4.2.5. Type 7 routes

Type 5 routes always get preference over Type 7 and the alternate ASBRs chosen for LFA calculation should belong to same type.Among Type 7 routes, routes with p-bit and forwarding address set have higher preference than routes without these attributes. Alternate ASBRs selected for LFA comparison should have same p-bit and forwarding address attributes.

4.2.6. Inequalities to be applied for alternate ASBR selection

The alternate ASBRs selected using above mechanism described in 3.2.1, are evaluated for Loop free criteria using below inequalities.

4.2.6.1. Forwarding address set to non zero value

```
  Link-Protection:
  F_opt(N,PO_i)+ cost(PO_i,P) < D_opt(N,S) +
                                F_opt(S,PO_best) + cost(PO_best,P)

  Link-Protection + Downstream-paths-only:
  F_opt(N,PO_i)+ cost(PO_i,P) < F_opt(S,PO_best) + cost(PO_best,P)

  Node-Protection:
  F_opt(N,PO_i)+ cost(PO_i,P) < D_opt(N,E) +
                                F_opt(E,PO_best) + cost(PO_best,P)

  Where,
  S             - The computing router
  N             - The alternate router being evaluated
  E             - The primary next-hop on shortest path from S to
                          prefix P.
  PO_i          - The specific prefix-originating router being
                          evaluated.
  PO_best       - The prefix-originating router on the shortest path
                          from the computing router S to prefix P.
  cost(X,Y)     - External cost for Y as advertised by X
  F_opt(X,Y)    - Distance on the shortest path from node X to Forwarding
                  address specified by ASBR Y.
  D_opt(X,Y)    - Distance on the shortest path from node X to node Y.
```

Figure 6: LFA inequality definition when forwarding address in non-zero

4.2.6.2. ASBRs advertising type1 and type2 cost

```
Link-Protection:
D_opt(N,PO_i)+ cost(PO_i,P) < D_opt(N,S) +
                              D_opt(S,PO_best) + cost(PO_best,P)

Link-Protection + Downstream-paths-only:
D_opt(N,PO_i)+ cost(PO_i,P) < D_opt(S,PO_best) + cost(PO_best,P)

Node-Protection:
D_opt(N,PO_i)+ cost(PO_i,P) < D_opt(N,E) +
                              D_opt(E,PO_best) + cost(PO_best,P)

Where,
S            - The computing router
N            - The alternate router being evaluated
E            - The primary next-hop on shortest path from S to
                        prefix P.
PO_i         - The specific prefix-originating router being
                        evaluated.
PO_best      - The prefix-originating router on the shortest path
                        from the computing router S to prefix P.
cost(X,Y)    - External cost for Y as advertised by X.
D_opt(X,Y)   - Distance on the shortest path from node X to node Y.
```

Figure 7: LFA inequality definition for type1 and type 2 cost

5. LFA Extended Procedures

This section explains the additional considerations in various aspects as listed below to the base LFA specification [RFC5286].

5.1. Links with IGP MAX_METRIC

Section 3.5 and 3.6 of [RFC5286] describes procedures for excluding nodes and links from use in alternate paths based on the maximum link metric (as defined in for IS-IS in [RFC5305] or as defined in [RFC3137] for OSPF). If these procedures are strictly followed, there are situations, as described below, where the only potential alternate available which satisfies the basic loop-free condition will not be considered as alternative.

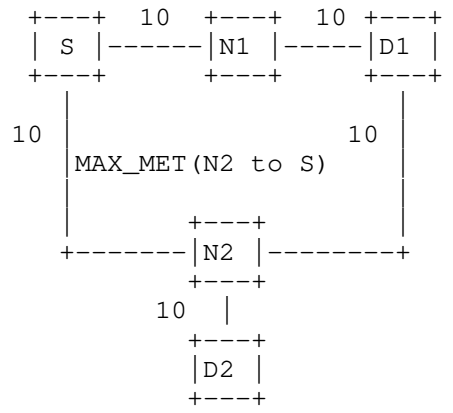

Figure 8: Link with IGP MAX_METRIC

In the simple example network, all the link costs have a cost of 10 in both directions, except for the link between S and N2. The S-N2 link has a cost of 10 in the direction from S to N2, and a cost of MAX_METRIC in the direction from N2 to S (0xffffff /2^24 - 1 for IS-IS and 0xffff for OSPF) for a specific end to end Traffic Engineering (TE) requirement of the operator. At node S, D1 is reachable through N1 with cost 20, and D2 is reachable through N2 with cost 20. Even though neighbor N2 satisfies basic loop-free condition (inequality 1 of [RFC5286]) for D1 this could be excluded as potential alternative because of the current exclusions as specified in section 3.5 and 3.6 procedure of [RFC5286]. But, as the primary traffic destined to D2 continue to use the link and hence irrespective of the reverse metric in this case, the same link MAY be used as a potential LFA for D1.

Alternatively, reverse metric of the link MAY be configured with MAX_METRIC-1, so that the link can be used as an alternative while meeting the TE requirements.

5.2. Multi Topology Considerations

Section 6.2 and 6.3.2 of [RFC5286] state that multi-topology OSPF and ISIS are out of scope for that specification. This memo clarifies and describes the applicability.

In Multi Topology (MT) IGP deployments, for each MT ID, a separate shortest path tree (SPT) is built with topology specific adjacencies, the LFA principles laid out in [RFC5286] are actually applicable for MT IS-IS [RFC5120] LFA SPF. The primary difference in this case is, identifying the eligible-set of neighbors for each LFA computation which is done per MT ID. The eligible-set for each MT ID is

determined by the presence of IGP adjacency from Source to the neighboring node on that MT-ID apart from the administrative restrictions and other checks laid out in [RFC5286]. The same is also applicable for OSPF [RFC4915] [MT-OSPF] or different AFs in multi instance OSPFv3 [RFC5838].

However for MT IS-IS, if a default topology is used with MT-ID 0 [RFC5286] and both IPv4 [RFC5305] and IPv6 routes/AFs [RFC5308] are present, then the condition of network congruency is applicable for LFA computation as well. Network congruency here refers to, having same address families provisioned on all the links and all the nodes of the network with MT-ID 0. Here with single decision process both IPv4 and IPv6 next-hops are computed for all the prefixes in the network and similarly with one LFA computation from all eligible neighbors per [RFC5286], all potential alternatives can be computed.

6. Acknowledgements

Thanks to Alia Atlas and Salih K A for their useful feedback and inputs.

7. IANA Considerations

N/A. - No protocol changes are proposed in this document.

8. Security Considerations

This document does not introduce any change in any of the protocol specifications and also this does not introduce any new security issues other than as noted in the LFA base specification [RFC5286].

9. References

9.1. Normative References

[RFC1195] Callon, R., "Use of OSI IS-IS for routing in TCP/IP and dual environments", RFC 1195, DOI 10.17487/RFC1195, December 1990, <http://www.rfc-editor.org/info/rfc1195>.

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

9.2. Informative References

[I-D.ietf-rtgwg-lfa-manageability]
Litkowski, S., Decraene, B., Filsfils, C., Raza, K., and M. Horneffer, "Operational management of Loop Free Alternates", draft-ietf-rtgwg-lfa-manageability-11 (work in progress), June 2015.

[RFC3137] Retana, A., Nguyen, L., White, R., Zinin, A., and D. McPherson, "OSPF Stub Router Advertisement", RFC 3137, DOI 10.17487/RFC3137, June 2001, <http://www.rfc-editor.org/info/rfc3137>.

[RFC4915] Psenak, P., Mirtorabi, S., Roy, A., Nguyen, L., and P. Pillay-Esnault, "Multi-Topology (MT) Routing in OSPF", RFC 4915, DOI 10.17487/RFC4915, June 2007, <http://www.rfc-editor.org/info/rfc4915>.

[RFC5120] Przygienda, T., Shen, N., and N. Sheth, "M-ISIS: Multi Topology (MT) Routing in Intermediate System to Intermediate Systems (IS-ISs)", RFC 5120, DOI 10.17487/RFC5120, February 2008, <http://www.rfc-editor.org/info/rfc5120>.

[RFC5286] Atlas, A., Ed. and A. Zinin, Ed., "Basic Specification for IP Fast Reroute: Loop-Free Alternates", RFC 5286, DOI 10.17487/RFC5286, September 2008, <http://www.rfc-editor.org/info/rfc5286>.

[RFC5305] Li, T. and H. Smit, "IS-IS Extensions for Traffic Engineering", RFC 5305, DOI 10.17487/RFC5305, October 2008, <http://www.rfc-editor.org/info/rfc5305>.

[RFC5308] Hopps, C., "Routing IPv6 with IS-IS", RFC 5308, DOI 10.17487/RFC5308, October 2008, <http://www.rfc-editor.org/info/rfc5308>.

[RFC5838] Lindem, A., Ed., Mirtorabi, S., Roy, A., Barnes, M., and R. Aggarwal, "Support of Address Families in OSPFv3", RFC 5838, DOI 10.17487/RFC5838, April 2010, <http://www.rfc-editor.org/info/rfc5838>.

Authors' Addresses

Pushpasis Sarkar (editor)
Individual

Email: pushpasis.ietf@gmail.com

Shraddha Hegde
Juniper Networks, Inc.
Electra, Exora Business Park
Bangalore, KA 560103
India

Email: shraddha@juniper.net

Chris Bowers
Juniper Networks, Inc.
1194 N. Mathilda Ave.
Sunnyvale, CA 94089
US

Email: cbowers@juniper.net

Uma Chunduri (editor)
Ericsson Inc.
300 Holger Way,
San Jose, California 95134
USA

Phone: 408 750-5678
Email: uma.chunduri@ericsson.com

Jeff Tantsura
Individual

Email: jefftant.ietf@gmail.com

Bruno Decraene
Orange

Email: bruno.decraene@orange.com

Hannes Gredler
Unaffiliated

Email: hannes@gredler.at

Network Working Group
Internet Draft

Category: Standard Track

L. Yong
W. Hao
D. Eastlake
Huawei
A. Qu
MetiaTek
J. Hudson
Brocade
U. Chunduri
Ericsson

Expires: May 2015 November 9, 2014

IGP Multicast Architecture

draft-yong-rtgwg-igp-multicast-arch-01

Abstract

This document specifies Interior Gateway Protocol (IGP) network architecture to support multicast transport. It describes the architecture components and the algorithms to automatically build a distribution tree for transporting multicast traffic and provides a method of pruning that tree for improved efficiency.

Status of this document

Table of Contents

1. Introduction

This document specifies Interior Gateway Protocol (IGP) network architecture to support multicast transport. It describes the architecture components and the algorithms to automatically build a distribution tree for transporting multicast traffic and provides a method of pruning that tree for improved efficiency.

An IGP network is built to transport unicast traffic. Traditionally, transporting multicast traffic relies on a protocol independent mechanism and a different protocol, i.e. PIM [RFC4601] [RFC5015]. The PIM protocol builds on top of IGP network and maintains its own states, which results longer convergence time for multicast traffic

Data Center infrastructure and advanced systems for cloud applications are looking for an IGP network to transport both unicast and multicast packets in a simpler and more efficient way than use of a separate protocol beyond IGP protocol. (see Section 1.1 for motivation)

This draft proposes the architecture and algorithms for an IGP based multicast transport. The architecture and algorithms automatically build a bi-directional distribution tree and pruned bi-directional tree for a multicast group without use of PIM. IGP protocol extension for this architecture is addressed in the [ISEXT].

1.1. Motivation

Network-as-a-service technically can be achieved by decoupling network IP space from service IP space as with a VxLAN [RFC7348] based network overlay. Decoupling network IP space from service IP address space also provides network agility and programmability to applications in a Data Center environment. To support all service applications, such IP network fabric must support both unicast and multicast. If network IP space is decoupled from service IP space, the network itself no longer needs manual configuration; automatically forming an IP network fabric can be done. The resulting "plug and play" can greatly simplify network operation.

With the goal of automation in forming a network fabric and support of any type of forwarding behavior the service applications require, IGP protocol should be extended to support:

1. Network formation

2. Multi destination distribution tree computation

Using external PIM prohibits the "automatic" nature requirement and results a longer convergence time of multicast transport than unicast transport because the convergence time for PIM is added to the basic IGP unicast route convergence time.

IGP based multicast reduces the number of protocols, states, and convergence time for multicast, which means a simpler underlay IP network that supports both unicast and multicast transport.

1.2. Conventions used in this document

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. IGP Architecture for Multicast Transport

An IGP multicast domain defined in this document contains edge routers and transit routers. Multicast source(s) and receiver(s) in a service space locally attach to edge routers or connect to edge routers through a layer 2 or layer 3 network that belongs to the same service space. When an ingress edge router receives a multicast packet from a multicast source in the service space, it replicates it along a pruned tree in the IGP domain. When an egress edge router receives a multicast packet from the IGP domain, it forwards the packet to the L2 or L3 service network that the receivers on and replicates the packet along the pruned tree in the domain. When a transit router receives a multicast packet from another router in the domain, it replicates the packet to its neighbor router(s) in the domain along a pruned tree.

An IGP multicast domain is used to carry L2 or L3 multicast traffic in a service (tenant) space in multi-tenant environment. Upon receiving a multicast packet from a source, the edge router first encapsulates the packet, adds its IP address as the source address and the corresponding underlay multicast IP address as the destination address on the encapsulated packet, then replicates it along a pruned tree. Egress edge router(s) decapsulate the packet before sending toward the receiver(s).

In an IGP multicast domain, each router has a unique IP address and the router IP address is advertised as a host address by IGP protocol. An IGP domain can be an IGP multicast domain if all

routers support the multicast capability described in this document; a subset of an IGP domain can be an IGP multicast domain where only some edge routers and transit routers have IGP multicast capability described in this draft and the draft [ISEXT]. In the case where the IGP multicast domain is subset of an IGP domain, a router in an IGP multicast domain must have at least one adjacency (next hop) to another router that is in the IGP multicast domain, that is, the IGP multicast domain must be connected. Configuring an IP tunnel between two routers in an IGP multicast domain can achieve this. How to configure such tunnel is outside the scope of this document.

In an IGP multicast domain, a default distribution tree is established automatically (see Section of 3.1). Operators may configure other distribution trees with different priorities in the domain as well and specify the associated multicast groups carried by these configured trees. By default, all the multicast groups use the default distribution tree.

The distribution tree computation algorithm is described in Section 3.2. The tree pruning for a particular multicast group is described in Section 3.3. Section 3.4 describes multiple trees to support one multicast group. Section 4 describes router forwarding procedures.

3. Computation Algorithms in IGP Multicast Domain

3.1. Automatic Tree Root Node Selection

By default the tree root is the router with the largest magnitude Router ID, considering the Router ID, i.e. router IPv4 address, to be an unsigned integer. Note that the algorithms in following sections use Router ID for router identifier, i.e. unique IP address assigned to a router in a IGP multicast domain.

Operators may configure a default tree root node (based on the topology) that takes precedence over the default tree root auto-calculated. This configured tree root node would advertise its IP address as the default tree root for all multicast groups that are not assigned to a distribution tree in a IGP multicast domain.

3.2. Distribution Tree Computation

The Distribution Tree Computation Algorithm uses the existing IGP Link State Database (LSDB). Based on the LSDB and shortest path algorithm, all routers in an IGP multicast domain calculate the distribution tree that has the default tree root node and reaches all the edge routers.

If an operator configures other distribution tree roots on other routers, the operator specifies what multicast groups use those trees and the tree root routers will advertise themselves as the tree root for those multicast groups by use of the new RTADDR TLV [ISEXT]. All routers in the domain will track the tree root nodes and calculate the path toward to each configured tree root node by using the shortest path algorithm, which form multiple distribute trees.

It is important that all the routers calculate the identical branches in a distribution tree in an IGP multicast domain. Section 3.2.1 and 3.3.2 specifies the tiebreaking rules for parent router selection in case of equal-cost path and for the link selection in case of multiple local links. Because link costs can be asymmetric, it is important for all tree construction calculations to use the cost towards the root.

3.2.1. Parent Selection

When there are equal costs from a potential child router to more than one possible parent router, all routers need to use the same tiebreakers. It is desirable to allow splitting traffic on as many links as possible in such situations when multiple distribution trees presents. This document uses the following tiebreaker rules:

If there are k distribution trees in the domain, when each router computes these trees, the k trees calculated are ordered and numbered from 0 to k-1 in ascending order according by root IP addresses.

The tiebreaker rule is: when building the tree number j, remember all possible equal cost parents for router N. After calculating the entire "tree" (actually, directed graph), for each router N, if N has "p" parents, then order the parents in ascending order according to the 7-octet IS-IS System ID considered as an unsigned integer, and number them starting at zero. For tree j, choose N's parent as choice (j-1) mod p.

3.2.2. Parallel Local Link Selection

If there are parallel point-to-point links between two routers, say R1 and R2, these parallel links would be visible to R1 and R2, but not to other routers. If this bundle of parallel links is included in a tree, it is important for R1 and R2 to decide which link to use; if the R1-R2 link is the branch for multiple trees, it is desirable to split traffic over as many links as possible. However the local link selection for a tree is irrelevant to other Routers.

Therefore, the tiebreaking algorithm need not be visible to any Routers other than R1 and R2.

When there are L parallel links between R1 and R2 and they both are on K trees. L links are ordered from 0 to L-1 in ascending order of Circuit ID as associated with the adjacency by the router with the highest System ID, and K trees are ordered from 0 to K-1 in ascending order of root IP addresses. The tiebreaker rule is: for tree k, select the link as choice k mod L.

Note that if multiple distribution trees are configured in a domain or on a router, better load balance among parallel links through the tie-breaking algorithm can be achieved. Otherwise, if there is only one tree is configured, then only one link in parallel links can be used for the corresponding distribution tree. However, calculating and maintaining many trees is resource consuming. Operators need to balance between two.

Another alternative is to use a lower level link aggregation protocol, such as [802.1AX-2011] on the parallel point-to-point links between R1 and R2. They will then appear to be a single link to the IGP and it will be the link aggregation protocol that spreads traffic across the actual lower level parallel links.

3.3. Multiple Distribute Trees for a Multicast Group

It is possible that a multicast group is associated with multiple trees that may have the same or different priority. When a multicast group associates with more than one tree, all routers have to select the same tree for the group. The tiebreaker rules specified in PIM [RFC4601] are used here. They are:

- Perform longest match on group-range to get a list of trees.
- Select the tree with highest priority.
- If only one tree with the highest priority, select the tree for the group-range.
- If multiple trees are with the highest priority, use the PIM hash function to choose one. PIM hash function is described in section 4.1.1 in RFC 4601 [RFC4601].

3.4. Pruning a Distribution Tree for a Group

Routers prune the distribution tree for each associated multicast group, i.e. eliminating branches that have no potential downstream

receivers. Multi-destination packets SHOULD only be forwarded on branches that are not pruned. The assumption here is that a multicast source is also a multicast receiver but a multicast receiver may not be a multicast source.

All routers in the domain receive LSP messages with GRADD-TLV [RFC7176] from the edge routers, which indicate which multicast group that an edge router is the receiver. According that, the routers prune the corresponding distribution tree for each multicast group and maintain a list of adjacency interfaces that are on the pruned tree for a multicast group. Among these interfaces, one interface will be toward the tree-root router (unless the router is the root) and zero or more interfaces will be toward some edge routers.

4. Router Forwarding Procedures

4.1. Packet Forwarding Along a Pruned Distribution Tree

Forwarding a multi-destination packet follows the pruned tree for the group that the packet belongs to. It is done as follows.

- If the router receives a multi-destination packet with group IP address that does not associated with any configured tree, the packet MUST be considered associated with the default tree.

- Else check if the link that the packet arrives on is one of the ports in the pruned distribution tree. If not, the packet MUST be dropped.

- Else optionally perform RPF checking (section 4.4). If the check is performed and it fails, the packet SHOULD be dropped.

- Else the packet is forwarded onto all the adjacency interfaces in the pruned tree for the group except the interface where the packet receive.

4.2. Local Forwarding at Edge Router

Upon receiving a multicast packet, besides forwarding it along the pruned tree, an edge router may also need to forward the packet to the local hosts attached to it. This is referred to as local forwarding in this document. Local forwarding table and multicast forwarding table in IGP domain should be stitched at each edge router. Local forwarding table can be generated using IGMP/PIM protocol running in the network between host and the edge router.

The local group database is needed to keep track of the group membership of attached hosts. Each entry in the local group database is a [group, host] pair, which indicates that the attached hosts belonging to the multicast group. When receiving a multicast packet, the edge router forwards the packet to the host that match the [group, host] pair in the local group database.

The local group database is built through the operation of the IGMPv3 [RFC3376]. An edge router sends periodic IGMPv3 Host Membership Queries to attached hosts. Hosts then respond with IGMPv3 Host Membership Reports, one for each multicast group to which they belong. Upon receiving a Host Membership Report for a multicast group A, the router updates its local group database by adding/refreshing the entry [group A, host] pair. If at a later time Reports for Group A cease to be heard from the host, the entry is then deleted from the local group database. The edge router further sends the LSP message with GRADDR TLV to inform other routers about the group memberships in the local group database.

4.2.1. Overlay Multicast Transport

An IGP multicast domain may be used to carry overlay multicast traffic. [RFC7365] There are two architecture scenarios:

1) IGP multicast domain edge router separates with overlay network edge device [RFC7365]. Before multicast traffic is forwarded, Overlay network should trigger underlay multicast domain to construct multicast tree using IGMP protocol in beforehand. Group address in the protocol is underlay multicast group address. Outer layer traffic encapsulation is performed on the overlay network edge device, IGP multicast domain acts as pure underlay network.

2) IGP multicast domain edge router collapses with overlay network edge device. Before multicast traffic is forwarded, local connecting host should trigger underlay multicast domain to construct multicast tree using IGMP like protocol beforehand. Group address in the protocol is overlay multicast group address, edge router should map the group address into underlay multicast group address.

The IGP multicast domain can support both scenarios. To carry overlay multicast traffic, a (designated) edge router (see Section below on Multi-Homing Access) further necessarily maintains the mapping between an overlay multicast group and a underlying multicast group, and performs packet encapsulation/descapsulation upon receiving a packet from a host or the underlay IGP network. Mapping between an overlay multicast group and a underlay multicast group can be manually configured, automatically generated by an

algorithm at a (designated) edge router. The same edge router MUST be selected as the Designated Forwarder for the overlay multicast group and underlying multicast group that are associated. If multiple overlay multicast groups attach to same edge router sets, these overlay multicast groups can be mapped to the same underlying multicast group to reduce underlay network multicast forwarding table size on each router. The mapping method is beyond the scope of this document.

4.3. Multi-homing Access Through Active-active MC-LAG

A multicast group receiver may attach to multiple edge routers through an active-active MC-LAG [802.1AX-2011] to enhance reliability.

When a remote edge router ingresses a multicast packet w/ multicast group address from local multicast source, if all egress routers in an MC-LAG forward the packet to the local host (receiver), the host will receive multiple copies of the multicast frame from the remote multicast source. To avoid duplicated packets received from the IGP domain to a local network, a Designated Forwarder (DF) mechanism is required. All the edge routers associated to a same MC-LAG use the same algorithm to select one DF edge router for a multicast group. Each MC-LAG should be assigned with a unique MC-LAG identifier in an IGP multicast domain, which may be manually configured or automatically provisioned. When an edge router in a MC-LAG receives a multicast group receiver joining message using IGMP/PIM like protocols, it announces its self MC-LAG ID and the multicast group correspondence to other routers in its IGP LSP. After network state reaches steady state, all edge routers in a MC-LAG elect the same router as DF for each multicast group. Upon receiving a multicast packet from the domain, only the DF edge router will forward the packet towards the receiver. All non-DF edge routers do not forward the packet towards the receiver.

All edge routers, including DF and non-DF, can ingress the traffic to IGP domain as usual. DF and non-DF state has influence only on the egress multicast traffic forwarding process.

If a multicast group source host attaches to multiple edge routers through an active-active MC-LAG, loop prevention, i.e. the packet sent by source host loops back to the source host via the edge routers in a MC-LAG, is necessary. The solutions for two scenarios are described below.

- When the multicast IGP domain edge routers separate with overlay network edge devices that carry overlay network traffic, these routers don't replace traffic source IP address when they inject the traffic into IGP domain. In this case, edge routers should acquire multicast source IP address in beforehand using a mechanism like IGMPv3 explicit tracking, and then the source IP addresses are synchronized among each edge routers in same MC-LAG. Then same split-horizon mechanism described in the above section can be used.

- When the multicast IGP domain edge routers collapse with overlay network edge devices, the edge router connecting to multicast source performs multicast encapsulation when it injects local multicast traffic into the IGP domain, source IP is the edge router's IP. Each edge router tracks the IP address(es) associated with the other edge router(s) with which it has shared MC-LAG. When the edge router receives a packet from an IGP domain, it examines the source IP address and filters out the packet on all local interfaces in the same MC-LAG. With this approach, local bias forwarding is required on the ingress edge router. It performs replication locally to all directly attached receivers no matter DF or non-DF state of the out interface connecting to each receiver.

4.4. Reverse Path Forwarding Check (RPFC)

The routing transients resulting from topology changes can cause temporary transient loops in distribution trees. If no precautions are taken, and there are fork points in such loops, it is possible for multiple copies of a packet to be forwarded. If this is a problem for a particular use, a Reverse Path Forwarding Check (RPFC) may be implemented.

In this case, the RPFC works by a router determining for each port, based on the source and destination IP address of a multicast packet, whether the port is a port that router expects to receive such a packet. In other words, is there an edge router with reachability to the source IP address such that, starting at that router and using the tree indicated by the destination IP address, the packet would have arrived at the port in question. If so, it is further distributed. If not, it is discarded. An RPFC can be implemented at some routers and not at others.

5. Security Considerations

To come in future version

6. IANA Considerations

This document does not request any IANA action.

7. Acknowledgements

Authors like to thank Mike McBride and Linda Dunbar for their valuable inputs.

8. References

8.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC2119, March 1997.

[RFC3376] Cain B., etc, "Internet Group Management Protocol, Version 3", rfc4604, October 2002

[RFC4601] Fenner, B., et al, "Protocol Independent multicast - Sparse Mode (PIM-SM): Protocol Specification", rfc4601, August 2006

[RFC5015] Handley, M., et al, "Bidirectional Protocol Independent Multicast (BIDIR-PIM", rfc5015, October 2007

[ISEXT] Yong, L., el al, "IS-IS Extension For Building Distribution Tree", draft-yong-isis-ext-4-distribution-tree, work in progress.

[802.1AX-2011] IEEE, "IEEE Standard for Local and metropolitan area networks - Link Aggregation", IEEE802.1AX, 2011

8.2. Informative References

[RFC7348] Mahalingam, M., Dutt, D., etc, "VXLAN: A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks", RFC7348, 2014

[RFC7365] Lasserre, M., "Framework for DC Network Virtualization", RFC7364, 2014.

Authors' Addresses

Lucy Yong
Huawei USA

Phone: 918-808-1918
Email: lucy.yong@huawei.com

Weiguo Hao
Huawei Technologies
101 Software Avenue,
Nanjing 210012
China

Phone: +86-25-56623144
Email: haoweiguo@huawei.com

Donald Eastlake
Huawei
155 Beaver Street
Milford, MA 01757 USA

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

Andrew Qu
MediaTek
San Jose, CA 95134 USA

Email: laodulaodu@gmail.com

Jon Hudson
Brocade
130 Holger Way
San Jose, CA 95134 USA

Phone: +1-408-333-4062
Email: jon.hudson@gmail.com

Uma Chunduri

Ericsson Inc.
300 Holger Way,
San Jose, California 95134
USA

Phone: 408-750-5678
Email: uma.chunduri@ericsson.com