RIFT WG
Internet-Draft
Intended status: Standards Track
Expires: December 13, 2019

T. Przygienda
Juniper Networks
Yuehua. Wei
Zheng. Zhang
ZTE Corporation
Dmitry. Afanasiev
Yandex
Tom. Verhaeg
Interconnect Services B.V.
Jaroslaw. Kowalczyk
Orange Polska

June 11, 2019

RIFT Applicability
draft-wei-rift-applicability-00

Abstract

This document discusses the properties and applicability of RIFT in different network topologies. It intends to provide a rough guide how RIFT can be deployed to simplify routing operations in Clos topologies and their variations.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

This document intends to explain the properties and applicability of RIFT [I-D.ietf-rift-rift] in different deployment scenarios and highlight the operational simplicity of the technology compared to traditional routing solutions.

2. Problem statement of a Fat Tree network in modern IP fabric

Clos and Fat-Tree topologies have gained prominence in today's networking, primarily as result of the paradigm shift towards a centralized data-center based architecture that is poised to deliver a majority of computation and storage services in the future.

Today's current routing protocols were geared towards a network with an irregular topology and low degree of connectivity originally. When they are applied to Fat-Tree topologies:

- There are always extensive configuration or provisioning during bring up and re-dimensioning.

- Both the spine node and the leaf node have the entire network topology and routing information, but in fact, the leaf node does not need so much complete information.

- There is significant Link State PDUs (LSPs) flooding duplication between spine nodes and leaf nodes during network bring up and topology update. It consumes both spine and leaf nodes' CPU and link bandwidth resources.

- When a spine node advertises a topology change, every leaf node connected to it will flood the update to all the other spine nodes, and those spine nodes will further flood them to all the leaf nodes, causing a O(n^2) flooding storm which is largely redundant.

3. Why ritf is chosen to address this use case

Further content of this document assumes that the reader is familiar with the terms and concepts used in OSPF [RFC2328] and IS-IS [ISO10589-Second-Edition] link-state protocols and at least the sections of RIFT [I-D.ietf-rift-rift] outlining the requirement of routing in IP fabrics and RIFT protocol concepts.

3.1. Overview of RIFT

RIFT is a dynamic routing protocol for Clos and fat-tree network topologies. It defines a link-state protocol when "pointing north" and path-vector protocol when "pointing south".

It floods flat link-state information northbound only so that each level obtains the full topology of levels south of it. That information is never flooded East-West or back South again. So a top tier node has full set of prefixes from the SPF calculation.

In the southbound direction the protocol operates like a "fully summarizing, unidirectional" path vector protocol or rather a distance vector with implicit split horizon whereas the information propagates one hop south and is 're-advertised' by nodes at next lower level, normally just the default route.

```
               +-----------+          +-----------+
               |    ToF    |          |    ToF    |         LEVEL 2
    +          +-----+--+--+          +-+--+------+
    |          |     |  |  |          | |  |      |      ^
    +          |     |  |  +-------------------------+   |
    Distance   |  +-------------------+ |  |      |  |   |
    Vector     |  |  |  |               |  |      |  |   +
    South      |  |  |  |      +--------+  |      |  |   Link+State
    +          |  |  |  |      |           |      |  |   Flooding
    |          |  |  +-------------+       |      |  |   North
    v          |  |     |      |   |       |      |  |   +
             +-+--+-+   +------+   +-------+   +--+--+-+ |
             |SPINE |   |SPINE |   | SPINE |   | SPINE | |  LEVEL 1
    +        ++----++   ++---+-+   +--+--+-+   ++----+-+ |
    +         |    |     |   |        |  |      |    |   |        ^N
    Distance  |    +-------+ |        |  +--------+  |   |        |   E
    Vector    |          | | |        |         | |  |   |     +------>
    South     |  +-------+ | |        | +-------+ |  |   |        |
    +         |  |         | |        | |         |  |   |        +
    v        ++--++      +-+-++      ++-+-+     +-+--++  +
             |LEAF|      |LEAF|      |LEAF|     |LEAF |     LEVEL 0
             +----+      +----+      +----+     +-----+
```

Figure 1: Rift overview

A middle tier node has only information necessary for its level, which are all destinations south of the node based on SPF calculation, default route and potential disaggregated routes.

RIFT combines the advantage of both Link-State and Distance Vector:

- Fastest Possible Convergence

- Automatic Detection of Topology

- Minimal Routes/Info on TORs

- High Degree of ECMP

- Fast De-commissioning of Nodes

- Maximum Propagation Speed with Flexible Prefixes in an Update

And RIFT eliminates the disadvantages of Link-State or Distance Vector:

- Reduced and Balanced Flooding

- Automatic Neighbor Detection

So there are two types of link state database which are "north representation" N-TIEs and "south representation" S-TIEs. The N-TIEs contain a link state topology description of lower levels and S-TIEs carry simply default routes for the lower levels.

There are a bunch of more advantages unique to RIFT listed below which could be understood if you read the details of RIFT [I-D.ietf-rift-rift].

- True ZTP

- Minimal Blast Radius on Failures

- Can Utilize All Paths Through Fabric Without Looping

- Automatic Disaggregation on Failures

- Simple Leaf Implementation that Can Scale Down to Servers

- Key-Value Store

- Horizontal Links Used for Protection Only

- Supports Non-Equal Cost Multipath and Can Replace MC-LAG

- Optimal Flooding Reduction and Load-Balancing

3.2. Applicable Topologies

Albeit RIFT is specified primarily for "proper" Clos or "fat-tree" structures, it already supports PoD concepts which are strictly speaking not found in original Clos concepts.

Further, the specification explains and supports operations of multi-plane Clos variants where the protocol relies on set of rings to allow the reconciliation of topology view of different planes as most desirable solution making proper disaggregation viable in case of failures. This observations hold not only in case of RIFT but in the generic case of dynamic routing on Clos variants with multiple planes and failures in bi-sectional bandwidth, especially on the leafs.

3.2.1. Horizontal Links

RIFT is not limited to pure Clos divided into PoD and multi-planes but supports horizontal links below the top of fabric level. Those links are used however only as routes of last resort when a spine

loses all northbound links or cannot compute a default route through them.

3.2.2. Vertical Shortcuts

Through relaxations of the specified adjacency forming rules RIFT implementations can be extended to support vertical "shortcuts" as proposed by e.g. [I-D.white-distoptflood]. The RIFT specification itself does not provide the exact details since the resulting solution suffers from either much larger blast radii with increased flooding volumes or in case of maximum aggregation routing bow-tie problems.

3.3. Use Cases

3.3.1. DC Fabrics

RIFT is largely driven by demands and hence ideally suited for application in underlay of data center IP fabrics, vast majority of which seem to be currently (and for the foreseeable future) Clos architectures. It significantly simplifies operation and deployment of such fabrics as described in Section 4 for environments compared to extensive proprietary provisioning and operational solutions.

3.3.2. Metro Fabrics

The demand for bandwidth is increasing steadily, driven primarily by environments close to content producers (server farms connection via DC fabrics) but in proximity to content consumers as well. Consumers are often clustered in metro areas with their own network architectures that can benefit from simplified, regular Clos structures and hence RIFT.

3.3.3. Building Cabling

Commercial edifices are often cabled in topologies that are either Clos or its isomorphic equivalents. With many floors the Clos can grow rather high and with that present a challenge for traditional routing protocols (except BGP and by now largely phased-out PNNI) which do not support an arbitrary number of levels which RIFT does naturally. Moreover, due to limited sizes of forwarding tables in active elements of building cabling the minimum FIB size RIFT maintains under normal conditions can prove particularly cost-effective in terms of hardware and operational costs.

3.3.4. Internal Router Switching Fabrics

It is common in high-speed communications switching and routing devices to use fabrics when a crossbar is not feasible due to cost, head-of-line blocking or size trade-offs. Normally such fabrics are not self-healing or rely on 1:/+1 protection schemes but it is conceivable to use RIFT to operate Clos fabrics that can deal effectively with interconnections or subsystem failures in such module. RIFT is neither IP specific and hence any link addressing connecting internal device subnets is conceivable.

3.3.5. CloudCO

The Cloud Central Office (CloudCO) is a new stage of telecom Central Office. It takes the advantage of Software Defined Networking (SDN) and Network Function Virtualization (NFV) in conjunction with general purpose hardware to optimize current networks. The following figure illustrates this architecture at a high level. It describes a single instance or macro-node of cloud CO. An Access I/O module faces a Cloud CO Access Node, and the CPEs behind it. A Network I/O module is facing the core network. The two I/O modules are interconnected by a leaf and spine fabric. [TR-384]

```
        +---------------------+           +----------------------+
        |         Spine       |           |     Spine            |
        |         Switch      |           |     Switch           |
        +------+---+------+-+-+           +--+-+-+-+-----+-------+
        |      |   |      | | |              | | | |     |       |
        |      |   |      | | +-------------------------------+  |
        |      |   |      | |                | | | |     |    |  |
        |      |   |      | +-------------------------+  |    |  |
        |      |   |      |                  | | | |  |  |    |  |
        |      |   +----------------------+  | | | |  |  |    |  |
        |      |          |               |  | | | |  |  |    |  |
        |  +---------------------------------+ | | |  |  |    |  |
        |  |   |          |               |    | | |  |  |    |  |
        |  |   |   +-----------------------------+ |  |  |    |  |
        |  |   |   |      |               |    |   |  |  |    |  |
        |  |   |   |      |   +--------------------+  |  |    |  |
        |  |   |   |      |   |           |    |      |  |    |  |
        |  |   |   |      |   |           |    |      |  |    |  |
        +--+ +-+---+--+ +-+---+--+     +--+----+--+ +-+--+--+ +--+
        |L | | Leaf   | | Leaf   |     |  Leaf    | | Leaf  | |L |
        |S | | Switch | | Switch |     |  Switch  | | Switch| |S |
        ++-+ +-+-+-+--+ +-+-+-+--+     +--+-+--+--+ ++-+--+-+ +-++
         |     | | |      | | |           | |  |     | |  |     |
         |   +-+-+-+--+ +-+-+-+--+     +--+-+--+--+ ++-+--+-+   |
         |   |Compute | |Compute |     | Compute  | |Compute|   |
         |   |Node    | |Node    |     | Node     | |Node   |   |
         |   |        | |        |     |          | |       |   |
         |   +--------+ +--------+     +----------+ +-------+   |
         |   || VAS5 || || vDHCP||     || vRouter|| ||VAS1 ||   |
         |   |--------| |--------|     |----------| |-------|   |
         |   |--------| |--------|     |----------| |-------|   |
         |   || VAS6 || || VAS3 ||     || v802.1x|| ||VAS2 ||   |
         |   |--------| |--------|     |----------| |-------|   |
         |   |--------| |--------|     |----------| |-------|   |
         |   || VAS7 || || VAS4 ||     ||  vIGMP || ||BAA  ||   |
         |   |--------| |--------|     |----------| |-------|   |
         |   +--------+ +--------+     +----------+ +-------+   |
         |                                                      |
        ++-----------+                                +---------++
        |Network I/O |                                |Access I/O|
        +------------+                                +----------+
```

Figure 2: An example of CloudCo architecture

The Spine-Leaf architectures deployed inside CloudCO meets the network requirements of adaptable, agile, scalable and dynamic.

4. Operational Simplifications and Considerations

RIFT presents the opportunity for organizations building and operating IP fabrics to simplify their operation and deployments while achieving many desirable properties of a dynamic routing on such a substrate:

- RIFT design follows minimum blast radius and minimum necessary epistemological scope philosophy which leads to very good scaling properties while delivering maximum reactiveness.

- RIFT allows for extensive Zero Touch Provisioning within the protocol. In its most extreme version RIFT does not rely on any specific addressing and for IP fabric can operate using IPv6 ND [RFC4861] only.

- RIFT has provisions to detect common IP fabric mis-cabling scenarios.

- RIFT negotiates automatically BFD per link allowing this way for IP and micro-BFD [RFC7130] to replace LAGs which do hide bandwidth imbalances in case of constituent failures. Further automatic link validation techniques similar to [RFC5357] could be supported as well.

- RIFT inherently solves many difficult problems associated with the use of traditional routing topologies with dense meshes and high degrees of ECMP by including automatic bandwidth balancing, flood reduction and automatic disaggregation on failures while providing maximum aggregation of prefixes in default scenarios.

- RIFT reduces FIB size towards the bottom of the IP fabric where most nodes reside and allows with that for cheaper hardware on the edges and introduction of modern IP fabric architectures that encompass e.g. server multi-homing.

- RIFT provides valley-free routing and with that is loop free. This allows the use of any such valley-free path in bi-sectional fabric bandwidth between two destination irrespective of their metrics which can be used to balance load on the fabric in different ways.

- RIFT includes a key-value distribution mechanism which allows for many future applications such as automatic provisioning of basic overlay services or automatic key roll-overs over whole fabrics.

- RIFT is designed for minimum delay in case of prefix mobility on the fabric.

- Many further operational and design points collected over many years of routing protocol deployments have been incorporated in RIFT such as fast flooding rates, protection of information lifetimes and operationally easily recognizable remote ends of links and node names.

4.1. Automatic Disaggregation

4.1.1. South reflection

South reflection is a mechanism that South Node TIEs are "reflected" back up north to allow nodes in same level without E-W links to "see" each other.

For example, Spine111\Spine112\Spine121\Spine122 reflects Node S-TIEs from ToF21 to ToF22 separately. Spine111\Spine112\Spine121\Spine122 reflects Node S-TIEs from ToF22 to ToF21 separately. So ToF22 and ToF21 knows each other as level 2 node.

As the result of the south reflection between Spine121-Leaf121-Spine122 and Spine121-Leaf122-Spine122, Spine121 and Spine 122 knows each other at level 1.

This is a use case to explain the deployment of a Fat-Tree and the algorithm to achieve automatic disaggregation.

4.1.2. Suboptimal routing upon link failure use case

```
                   +--------+          +--------+
                   |        |          |        |
                   | ToF21  |          |  ToF22 |                LEVEL 2
                   ++-+--+-++          ++-+--+-++
                    | |  | |            | |  | |
                    | |  | |            | |  | linkTS8
                    | |  | |            | |  | |
                    | |  | |            | |  | |
     +--------------+ |  +--linkTS3--+  | |  | +--------------+
     |                |    |         |  | |  |                |
     |    +-----------------------------+ |  linkTS7          |
     |    |           |    |         |    |  |                |
     |    |           |    +--------linkTS4-------------+     |
     |    |           |              |    |  |          |     |
     |    |           +-+    +---------------+          |     |
     |    |             |    |       |  linkTS6         |     |
   +-+----++          +-+-----+     ++----+-+          ++-----++
   |       |          |       |     |       |          |       |
   |Spin111|          |Spin112|     |Spin121|          |Spin122| LEVEL 1
   +-+---+-+          ++----+-+     +-+---+-+          ++---+--+
     |   |             |    |         |   |             |   |
     |   +---------------+  |         |   +-XX-linkSL6----+ |
     |                 | |  |      linkSL5              | | linkSL8
     |   +-------------+ |  |         |   +----linkSL7--+ | |
     |   |               |  |         |   |               | |
   +-+---+-+          +--+--+-+     +-+---+-+          +--+-+--+
   |       |          |       |     |       |          |       |
   |Leaf111|          |Leaf112|     |Leaf121|          |Leaf122| LEVEL 0
   +-+-----+          ++------+     +-----+-+          +-+-----+
     +                 +                  +              +
   Prefix111          Prefix112     Prefix121          Prefix122
```

Figure 3: Suboptimal routing upon link failure use case

As shown in figure above, as the result of the south reflection between Spine121-Leaf121-Spine122 and Spine121-Leaf122-Spine122, Spine121 and Spine 122 knows each other at level 1.

Without disaggregation mechanism, when linkSL6 fails, the packet from leaf121 to prefix122 will probably go up through linkSL5 to linkTS3 then go down through linkTS4 to linkSL8 to Leaf122 or go up through linkSL5 to linkTS6 then go down through linkTS4 and linkSL8 to Leaf122 based on pure default route. It's the case of suboptimal routing.

With disaggregation mechanism, when linkSL6 fails, Spine122 will detect the failure according to the reflected node S-TIE from Spine121. Based on the disaggregation algorithm provided by RITF,

Spine122 will explicitly advertise prefix122 in Prefix S-TIE SouthPrefixesElement(prefix122, cost 1). The packet from leaf121 to prefix122 will only be sent to linkSL7 following a longest-prefix match to prefix 122 directly then go down through linkSL8 to Leaf122 .

4.1.3. Black-holing upon link failure use case

```
                   +--------+          +--------+
                   |        |          |        |
                   | ToF 21 |          | ToF 22 |                LEVEL 2
                   ++-+--+-++          ++-+--+-++
                    | |  | |            | |  | |
                    | |  | |            | |  | linkTS8
                    | |  | |            | |  | |
                    | |  | |            | |  | |
     +--------------+ |  +--linkTS3-X+  | |  | +--------------+
     linkTS1          |    |         |  | |  |                |
     |    +-----------------------------+ |  linkTS7          |
     |    |           |    |         |    |  |                |
     |    |      linkTS2   +--------linkTS4-X-----------+     |
     |    |           |              |    |  |          |     |
     |   linkTS5      +-+    +---------------+          |     |
     |    |             |    |       |  linkTS6         |     |
   +-+----++          +-+-----+     ++----+-+          ++-----++
   |       |          |       |     |       |          |       |
   |Spin111|          |Spin112|     |Spin121|          |Spin122| LEVEL 1
   +-+---+-+          ++----+-+     +-+---+-+          ++---+--+
     |   |             |    |         |   |             |   |
     |   +---------------+  |         |   +----linkSL6----+ |
     linkSL1           | |  |      linkSL5              | | linkSL8
     |   +---linkSL3---+ |  |         |   +----linkSL7--+ | |
     |   |               |  |         |   |               | |
   +-+---+-+          +--+--+-+     +-+---+-+          +--+-+--+
   |       |          |       |     |       |          |       |
   |Leaf111|          |Leaf112|     |Leaf121|          |Leaf122| LEVEL 0
   +-+-----+          ++------+     +-----+-+          +-+-----+
     +                 +                  +              +
   Prefix111          Prefix112     Prefix121          Prefix122
```

Figure 4: Black-holing upon link failure use case

This scenario illustrates a case when double link failure occurs, black-holing happens.

Without disaggregation mechanism, when linkTS3 and linkTS4 both fail, the packet from leaf111 to prefix122 would suffer 50% black-holing based on pure default route. The packet supposed to go up through

linkSL1 to linkTS1 then go down through linkTS3 or linkTS4 will be dropped. The packet supposed to go up through linkSL3 to linkTS2 then go down through linkTS3 or linkTS4 will be dropped as well. It's the case of black-holing.

With disaggregation mechanism, when linkTS3 and linkTS4 both fail, ToF22 will detect the failure according to the reflected node S-TIE of ToF21 from Spine111\Spine112\Spine121\Spine122. Based on the disaggregation algorithm provided by RITF, ToF22 will explicitly originate an S-TIE with prefix 121 and prefix 122, that is flooded to spines 111, 112, 121 and 122.

The packet from leaf111 to prefix122 will not be routed to linkTS1 or linkTS2. The packet from leaf111 to prefix122 will only be routed to linkTS5 or linkTS7 following a longest-prefix match to prefix122.

4.2. Usage of ZTP

Each RIFT node may operate in zero touch provisioning (ZTP) mode. It has no configuration (unless it is a Top-of-Fabric at the top of the topology or the must operate in the topology as leaf and/or support leaf-2-leaf procedures) and it will fully configure itself after being attached to the topology.

The most import component for ZTP is the automatic level derivation procedure. All the Top-of-Fabric nodes are explicitly marked with TOP_OF_FABRIC flag which are initial 'seeds' needed for other ZTP nodes to derive their level in the topology.

The derivation of the level of each node happens based on LIEs received from its neighbors whereas each node (with possibly exceptions of configured leafs) tries to attach at the highest possible point in the fabric.

This guarantees that even if the diffusion front reaches a node from "below" faster than from "above", it will greedily abandon already negotiated level derived from nodes topologically below it and properly peers with nodes above.

5. Normative References

[I-D.ietf-rift-rift]
Team, T., "RIFT: Routing in Fat Trees", draft-ietf-rift-rift-05 (work in progress), April 2019.

[I-D.white-distoptflood]
White, R. and S. Zandi, "IS-IS Optimal Distributed Flooding for Dense Topologies", draft-white-distoptflood-00 (work in progress), March 2019.

[ISO10589-Second-Edition]
International Organization for Standardization, "Intermediate system to Intermediate system intra-domain routeing information exchange protocol for use in conjunction with the protocol for providing the connectionless-mode Network Service (ISO 8473)", Nov 2002.

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

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

[RFC5357] Hedayat, K., Krzanowski, R., Morton, A., Yum, K., and J. Babiarz, "A Two-Way Active Measurement Protocol (TWAMP)", RFC 5357, DOI 10.17487/RFC5357, October 2008, <https://www.rfc-editor.org/info/rfc5357>.

[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, <https://www.rfc-editor.org/info/rfc7130>.

[TR-384] Broadband Forum Technical Report, "TR-384 Cloud Central Office Reference Architectural Framework", Jan 2018.

Authors' Addresses

Tony Przygienda
Juniper Networks
1194 N. Mathilda Ave
Sunnyvale, CA 94089
US

Email: prz@juniper.net

Yuehua Wei
ZTE Corporation
No.50, Software Avenue
Nanjing 210012
P. R. China

Email: wei.yuehua@zte.com.cn

Zheng Zhang
ZTE Corporation
No.50, Software Avenue
Nanjing 210012
P. R. China

Email: zzhang_ietf@hotmail.com

Dmitry Afanasiev
Yandex

Email: fl0w@yandex-team.ru

Tom Verhaeg
Interconnect Services B.V.

Email: t.verhaeg@interconnect.nl

Jaroslaw Kowalczyk
Orange Polska

Email: jaroslaw.kowalczyk2@orange.com