Stream: Internet Engineering Task Force (IETF)
RFC: 9696
Category: Informational
Published: December 2024
ISSN: 2070-1721
Authors: Y. Wei, Ed. *ZTE Corporation*
Z. Zhang *ZTE Corporation*
D. Afanasiev *Yandex*
P. Thubert *Individual*
T. Przygienda *Juniper Networks*

RFC 9696
Routing in Fat Trees (RIFT) Applicability and Operational Considerations

Abstract

This document discusses the properties, applicability, and operational considerations of Routing in Fat Trees (RIFT) in different network scenarios with the intention of providing a rough guide on how RIFT can be deployed to simplify routing operations in Clos topologies and their variations.

Status of This Memo

This document is not an Internet Standards Track specification; it is published for informational purposes.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Not all documents approved by the IESG are candidates for any level of Internet Standard; see Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc9696.

Copyright Notice

Table of Contents

1. Introduction

This document discusses the properties and applicability of "RIFT: Routing in Fat Trees" [RFC9692] in different deployment scenarios and highlights the operational simplicity of the technology compared to classical routing solutions. It also documents special considerations when RIFT is used with or without overlays and/or controllers and how RIFT identifies miscablings and reroutes around node and link failures.

2. Terminology

This document uses the terminology defined in [RFC9692]. The most frequently used terms and their definitions from that document are listed here.

Clos / Fat Tree:
: This document uses the terms "Clos" and "Fat Tree" interchangeably where it always refers to a folded spine-and-leaf topology with possibly multiple Points of Delivery (PoDs) and one or multiple Top of Fabric (ToF) planes. Several modifications such as leaf-2-leaf shortcuts and multiple level shortcuts are possible and described further in the document.

Crossbar:
: Physical arrangement of ports in a switching matrix without implying any further scheduling or buffering disciplines.

Directed Acyclic Graph (DAG):
: A finite directed graph with no directed cycles (loops). If links in a Clos are considered as either being all directed towards the top or bottom, each of such two graphs is a DAG.

Disaggregation:
: The process in which a node decides to advertise more specific prefixes southwards, either positively to attract the corresponding traffic or negatively to repel it. Disaggregation is performed to prevent traffic loss and suboptimal routing to the more specific prefixes.

Leaf:
: A node without southbound adjacencies. Level 0 implies a leaf in RIFT, but a leaf does not have to be level 0.

LIE:
: This is an acronym for "Link Information Element" exchanged on all the system's links running RIFT to form *ThreeWay* adjacencies and carry information used to perform RIFT Zero Touch Provisioning (ZTP) of levels.

South Reflection:
: Often abbreviated just as "reflection", South Reflection defines a mechanism where South Node TIEs are "reflected" from the level south back up north to allow nodes in the same level without East-West links to be aware of each other's node Topology Information Elements (TIEs).

Spine:
: Any nodes north of leaves and south of ToF nodes. Multiple layers of spines in a PoD are possible.

TIE:
: This is an acronym for "Topology Information Element". TIEs are exchanged between RIFT nodes to describe parts of a network such as links and address prefixes. A TIE always has a direction and a type. North TIEs (sometimes abbreviated as N-TIEs) are used when dealing

with TIEs in the northbound representation, and South-TIEs (sometimes abbreviated as S-TIEs) are used for the southbound equivalent. TIEs have different types, such as node and prefix TIEs.

3. Problem Statement of Routing in Modern IP Fabric Fat Tree Networks

Clos [CLOS] topologies (commonly called a Fat Tree/network in modern IP fabric considerations as a similar term for the original definition of the term Fat Tree [FATTREE]) have gained prominence in today's networking, primarily as a result of the paradigm shift towards a centralized data-center-based architecture that delivers a majority of computation and storage services.

Current routing protocols were geared towards a network with an irregular topology with isotropic properties and a low degree of connectivity. When applied to Fat Tree topologies:

- They tend to need extensive configuration or provisioning during initialization and adding or removing nodes from the fabric.
- For link-state routing protocols, all nodes including spine-and-leaf nodes learn the entire network topology and routing information, which is actually not needed on the leaf nodes during normal operation. They flood significant amounts of duplicate link-state information between spine-and-leaf nodes during topology updates and convergence events, requiring that additional CPU and link bandwidth be consumed. This may impact the stability and scalability of the fabric, make the fabric less reactive to failures, and prevent the use of cheaper hardware at the lower levels (i.e., spine-and-leaf nodes).

4. Applicability of RIFT to Clos IP Fabrics

Further content of this document assumes that the reader is familiar with the terms and concepts used in the Open Shortest Path First (OSPF) [RFC2328], OSPF for IPv6 [RFC5340], and Intermediate System to Intermediate System (IS-IS) [ISO10589-Second-Edition] link-state protocols. [RFC9692] outlines the requirements of routing in IP fabrics and RIFT protocol concepts.

4.1. Overview of RIFT

RIFT is a dynamic routing protocol that is tailored for use in Clos, Fat Tree, and other anisotropic topologies. Therefore, a core property of RIFT is that its operation is sensitive to the structure of the fabric -- it is anisotropic. RIFT acts as a link-state protocol when "pointing north", advertising southward routes to northward peers (parents) through flooding and database synchronization. When "pointing south", RIFT operates hop-by-hop like a distance-vector protocol, typically advertising a fabric default route towards the ToF, aka superspine, to southward peers (children).

The fabric default is typically the default route as described in Section 6.3.8 ("Southbound Default Route Origination") of [RFC9692]. The ToF nodes may alternatively originate more specific prefixes (P') southbound instead of the default route. In such a scenario, all addresses carried within the RIFT domain must be contained within P', and it is possible for a leaf that acts as gateway to the Internet to advertise the default route instead.

RIFT floods flat link-state information northbound only so that each level obtains the full topology of the levels that are south of it. That information is never flooded East-West or back south again, so a top tier node has a full set of prefixes from the Shortest Path First (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. Routing information, normally just the default route, propagates one hop south and is "re-advertised" by nodes at next lower level.

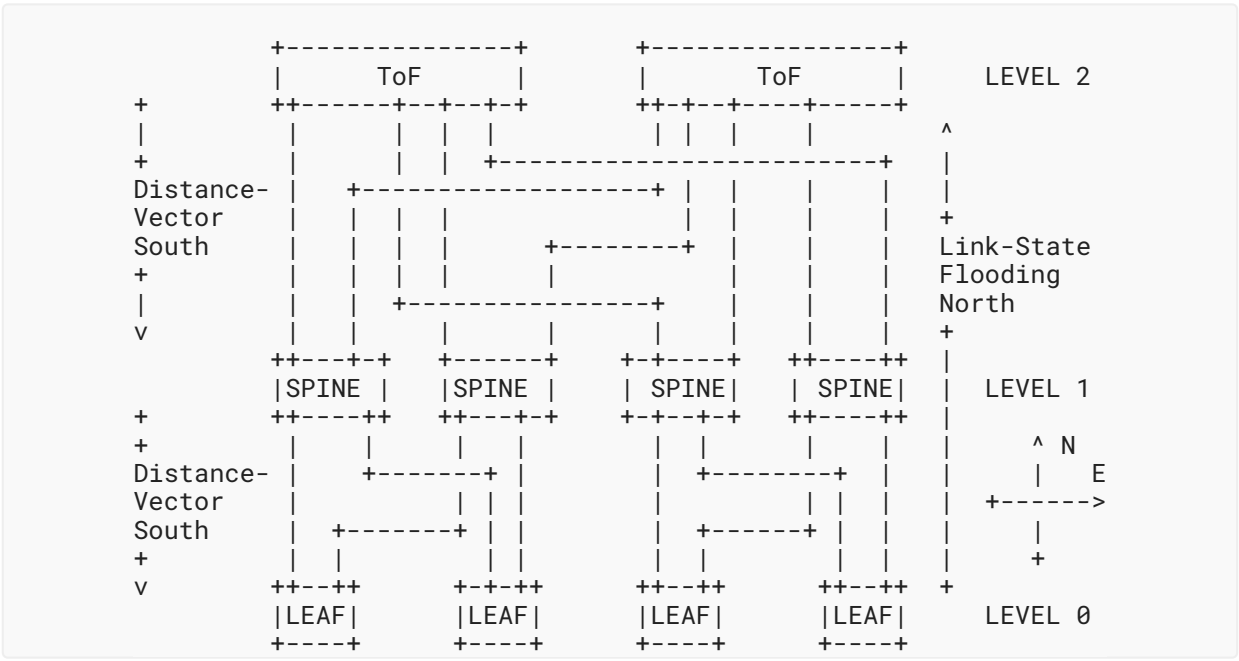

Figure 1: RIFT Overview

A spine node only has information necessary for its level, which is all destinations south of the node based on SPF calculation, the default route, and potentially disaggregated routes.

RIFT combines the advantages of both link-state and distance-vector protocols:

- Fastest possible convergence
- Automatic detection of topology
- Minimal routes/information on Top-of-Rack (ToR) switches, aka leaf nodes
- High degree of ECMP
- Fast decommissioning of nodes

- Maximum propagation speed with flexible prefixes in an update

There are two types of link-state databases that are "north representation" North Topology Information Elements (N-TIEs) and "south representation" South Topology Information Elements (S-TIEs). The N-TIEs contain a link-state topology description of lower levels, and the S-TIEs simply carry default and disaggregated routes for the lower levels.

RIFT also eliminates major disadvantages of link-state and distance-vector protocols with the following:

- Reduced and balanced flooding
- Level-constrained automatic neighbor discovery

To achieve this, RIFT builds on the art of IGPs, such as OSPF, IS-IS, Mobile Ad Hoc Network (MANET), and Internet of Things (IoT) to provide unique features:

- Automatic (positive or negative) route disaggregation of northward routes upon fallen leaves
- Recursive operation in the case of negative route disaggregation
- Anisotropic routing that extends a principle seen in the Routing Protocol for Low-Power and Lossy Networks (RPL) [RFC6550] to wide superspines
- Optimal flooding reduction that derives from the concept of a "multipoint relay" (MPR) found in Optimized Link State Routing (OLSR) [RFC3626] and balances the flooding load over northbound links and nodes

Additional advantages that are unique to RIFT are listed below. The details of these advantages can be found in RIFT [RFC9692].

- True ZTP
- Minimal blast radius on failures
- Can utilize all paths through fabric without looping
- Simple leaf implementation that can scale down to servers
- Key-value store
- Horizontal links used for protection only

4.2. Applicable Topologies

Albeit RIFT is specified primarily for "proper" Clos or Fat Tree topologies, the protocol natively supports Points of Delivery (PoD) concepts, which, strictly speaking, are not found in the original Clos concept.

Further, the specification explains and supports operations of multi-plane Clos variants where the protocol recommends the use of inter-plane rings at the ToF level to allow the reconciliation of topology view of different planes to make the Negative Disaggregation viable in case of failures within a plane. These observations hold not only in case of RIFT but also in the generic case of dynamic routing on Clos variants with multiple planes and failures in bisectional bandwidth, especially on the leaves.

4.2.1. Horizontal Links

RIFT is not limited to pure Clos divided into PoD and multi-planes but supports horizontal (East-West) links below the ToF level. Those links are used only for last resort northbound forwarding when a spine loses all its northbound links or cannot compute a default route through them.

A full-mesh connectivity between nodes on the same level can be deployed, which allows North SPF (N-SPF) to provide for any node losing all its northbound adjacencies (as long as any of the other nodes in the level are northbound connected) and still participate in northbound forwarding.

Note that a "ring" of horizontal links at any level below ToF does not provide a "ring-based protection" scheme since the SPF computation would have to deal with breaking of "loops", an application for which RIFT is not intended.

4.2.2. Vertical Shortcuts

Through relaxations of the specified adjacency forming rules, RIFT implementations can be extended to support vertical "shortcuts". The RIFT specification itself does not provide the exact details since the resulting solution suffers from either a much larger blast radius with increased flooding volumes or bow tie problems in the case of maximum aggregation routing.

4.2.3. Generalizing to Any Directed Acyclic Graph

RIFT is an anisotropic routing protocol, meaning that it has a sense of direction (northbound, southbound, and East-West) and operates differently depending on the direction.

Since a DAG provides a sense of north (the direction of the DAG) and south (the reverse), it can be used to apply RIFT -- an edge in the DAG that has only incoming vertices is a ToF node.

There are a number of caveats though:

- The DAG structure must exist before RIFT starts, so there is a need for a companion protocol to establish the logical DAG structure.
- A generic DAG does not have a sense of East and West. The operation specified for East-West links and the southbound reflection between nodes are not applicable. Also, ZTP will derive a sense of depth that will eliminate some links. Variations of ZTP could be derived to meet specific objectives, e.g., make it so that most routers have at least two parents to reach the ToF.
- RIFT applies to any Destination-Oriented DAG (DODAG) where there's only one ToF node and the problem of disaggregation does not exist. In that case, RIFT operates very much like RPL [RFC6550], but uses link-state information for southbound routes (downwards in RPL's terms). For an arbitrary DAG with multiple destinations (ToFs), the way disaggregation happens has to be considered.
- Positive Disaggregation expects that most of the ToF nodes reach most of the leaves, so disaggregation is the exception as opposed to the rule. When this is no longer true, it makes sense to turn off disaggregation and route between the ToF nodes over a ring, a full mesh, a

transit network, or a form of area zero. Then again, this operation is similar to RPL operating as a single DODAG with a virtual root.

- In order to aggregate and disaggregate routes, RIFT requires that all the ToF nodes share the full knowledge of the prefixes in the fabric. This can be achieved with a ring as suggested by RIFT [RFC9692], by some preconfiguration, or by using a synchronization with a common repository where all the active prefixes are registered.

4.2.4. Reachability of Internal Nodes in the Fabric

RIFT does not require that nodes have reachable addresses in the fabric, though it is clearly desirable for operational purposes. Under normal operating conditions, this can be easily achieved by injecting the node's loopback address into Prefix North TIEs and Prefix South TIEs or other implementation-specific mechanisms.

Special considerations arise when a node loses all northbound adjacencies but is not at the top of the fabric. If a spine node loses all northbound links, the spine node doesn't advertise a default route. But if the level of the spine node is auto-determined by ZTP, it will "fall down" as depicted in Figure 8.

4.3. Use Cases

4.3.1. Data Center Topologies

4.3.1.1. Data Center Fabrics

RIFT is suited for applying underlay routing in data center (DC) IP fabrics, with the vast majority of these IP fabrics being Clos architectures (and will be for the foreseeable future). It significantly simplifies operation and deployment of such fabrics as described in Section 5 for environments compared to extensive proprietary provisioning and operational solutions.

4.3.1.2. Adaptations to Other Proposed Data Center Topologies

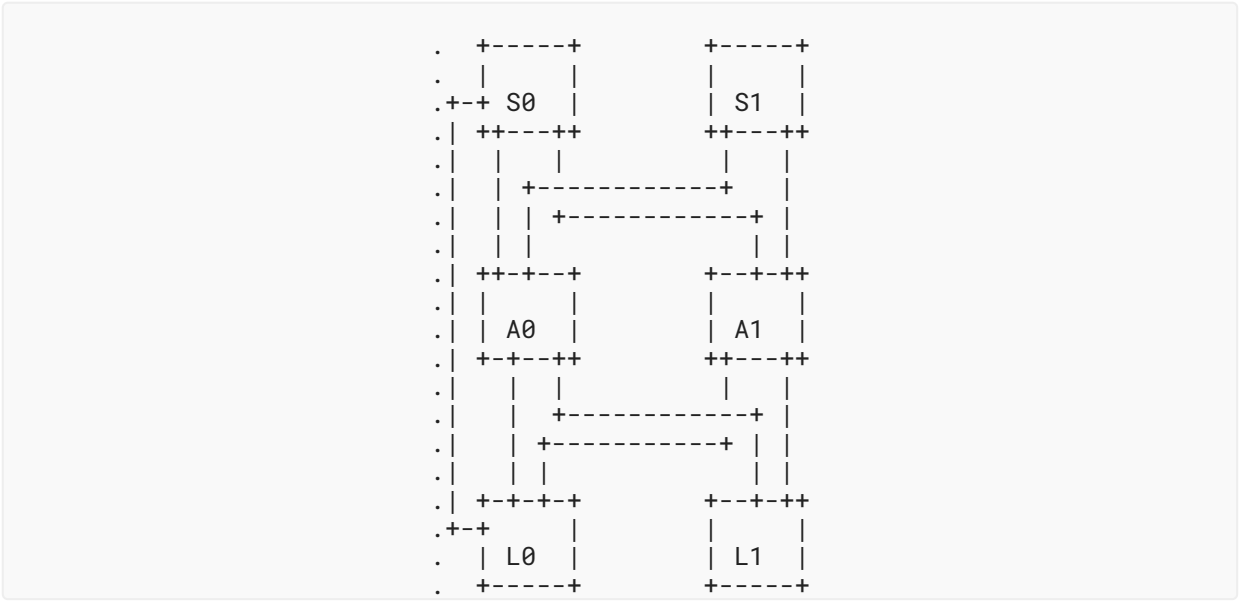

Figure 2: Level Shortcut

RIFT is not strictly limited to Clos topologies. The protocol only requires a sense of "compass rose directionality" either achieved through configuration or derivation of levels. So conceptually, shortcuts between levels could be included. Figure 2 depicts an example of a shortcut between levels. In this example, suboptimal routing will occur when traffic is sent from L0 to L1 via S0's default route and back down through A0 or A1. In order to avoid that, only default routes from A0 or A1 are used. All leaves would be required to install each other's routes.

While various technical and operational challenges may require the use of such modifications, discussion of those topics is outside the scope of this document.

4.3.2. Metro Networks

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. Thus, they can also benefit from RIFT.

4.3.3. Building Cabling

Commercial edifices are often cabled in topologies that are either Clos or its isomorphic equivalents. The Clos can grow rather high with many levels. That presents a challenge for classical routing protocols (except BGP [RFC4271] and Private Network-Network Interface (PNNI) [PNNI], which is largely phased-out by now) that do not support an arbitrary number of levels, which RIFT does naturally. Moreover, due to the limited sizes of forwarding tables in network elements of building cabling, the minimum FIB size RIFT maintains under normal conditions is cost-effective in terms of hardware and operational costs.

4.3.4. Internal Router Switching Fabrics

It is common in high-speed communications switching and routing devices to use switch fabrics that are interconnection networks inside the devices connecting the input ports to their output ports. For example, a crossbar is one of the switch fabric techniques, even though it 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 or 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 a module. RIFT is not IP specific and hence any link addressing connecting internal device subnets is conceivable.

4.3.5. CloudCO

The Cloud Central Office (CloudCO) is a new stage of the 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 CloudCO that provides a number of value-added services (VASes), a Broadband Access Abstraction (BAA), and virtualized network services. An Access I/O module faces a CloudCO access node and the Customer Premises Equipment (CPE) behind it. A Network I/O module is facing the core network. The two I/O modules are interconnected by a spine-and-leaf fabric [TR-384].

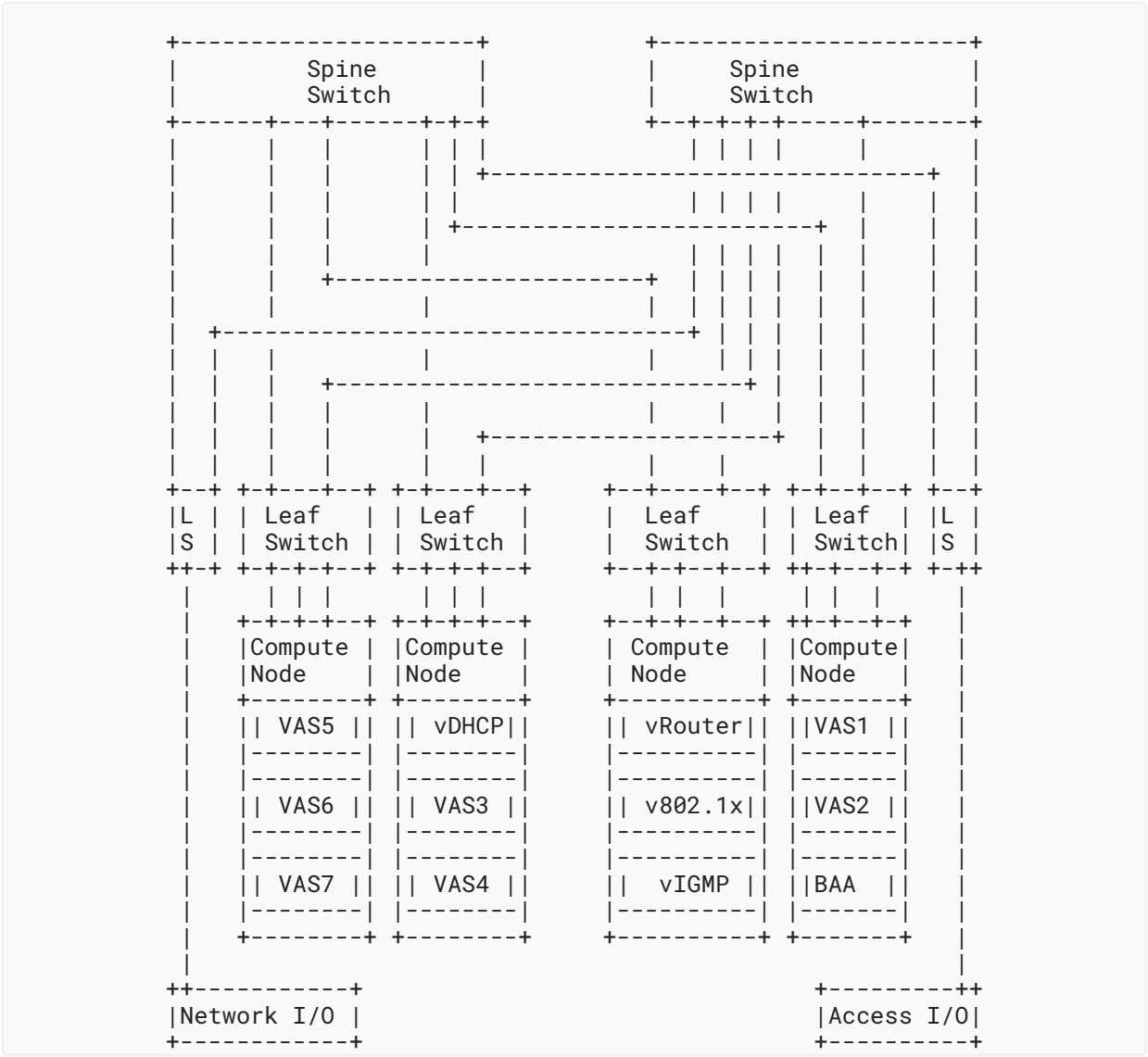

Figure 3: CloudCO Architecture Example

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

5. Operational Considerations

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

- RIFT only floods routing information to the devices that need it.

- RIFT allows for ZTP within the protocol. In its most extreme version, RIFT does not rely on any specific addressing and can operate using IPv6 Neighbor Discovery (ND) [RFC4861] only for IP fabric.
- RIFT has provisions to detect common IP fabric miscabling scenarios.
- RIFT automatically negotiates Bidirectional Forwarding Detection (BFD) per link. This allows for IP and micro-BFD [RFC7130] to replace Link Aggregation Groups (LAGs) that hide bandwidth imbalances in case of constituent failures. Further automatic link validation techniques similar to those in [RFC5357] could be supported as well.
- RIFT inherently solves many problems associated with the use of classical 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. ECMP in RIFT eliminates the need for more Loop-Free Alternate (LFA) procedures.
- RIFT reduces FIB size towards the bottom of the IP fabric where most nodes reside. This allows for cheaper hardware on the edges and introduction of modern IP fabric architectures that encompass server multihoming and other mechanisms.
- RIFT provides valley-free routing that is loop free. A valley-free path allows for reversal of direction at most once from a packet heading northbound to southbound while permitting traversal of horizontal links in the northbound phase. This allows for the use of any such valley-free path in bisectional fabric bandwidth between two destinations irrespective of their metrics that can be used to balance load on the fabric in different ways. Valley-free routing eliminates the need for any specific micro-loop avoidance procedures for RIFT.
- RIFT includes a key-value distribution mechanism that allows for future applications such as automatic provisioning of basic overlay services or automatic key rollovers over whole fabrics.
- RIFT is designed for minimum delay in case of prefix mobility on the fabric. In conjunction with [RFC8505], RIFT can differentiate anycast advertisements from mobility events and retain only the most recent advertisement in the latter case.
- 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 recognizable remote ends of links and node names.

5.1. South Reflection

South reflection is a mechanism where South Node TIEs are "reflected" back up north to allow nodes in the same level without East-West links to "see" each other.

For example, in Figure 4, Spine111\Spine112\Spine121\Spine122 reflects Node S-TIEs from ToF21 to ToF22 separately. Respectively, Spine111\Spine112\Spine121\Spine122 reflects Node S-TIEs from ToF22 to ToF21 separately, so ToF22 and ToF21 see each other's node information as level 2 nodes.

In an equivalent fashion, as the result of the south reflection between Spine121-Leaf121-Spine122 and Spine121-Leaf122-Spine122, Spine121 and Spine 122 know each other at level 1.

5.2. Suboptimal Routing on Link Failures

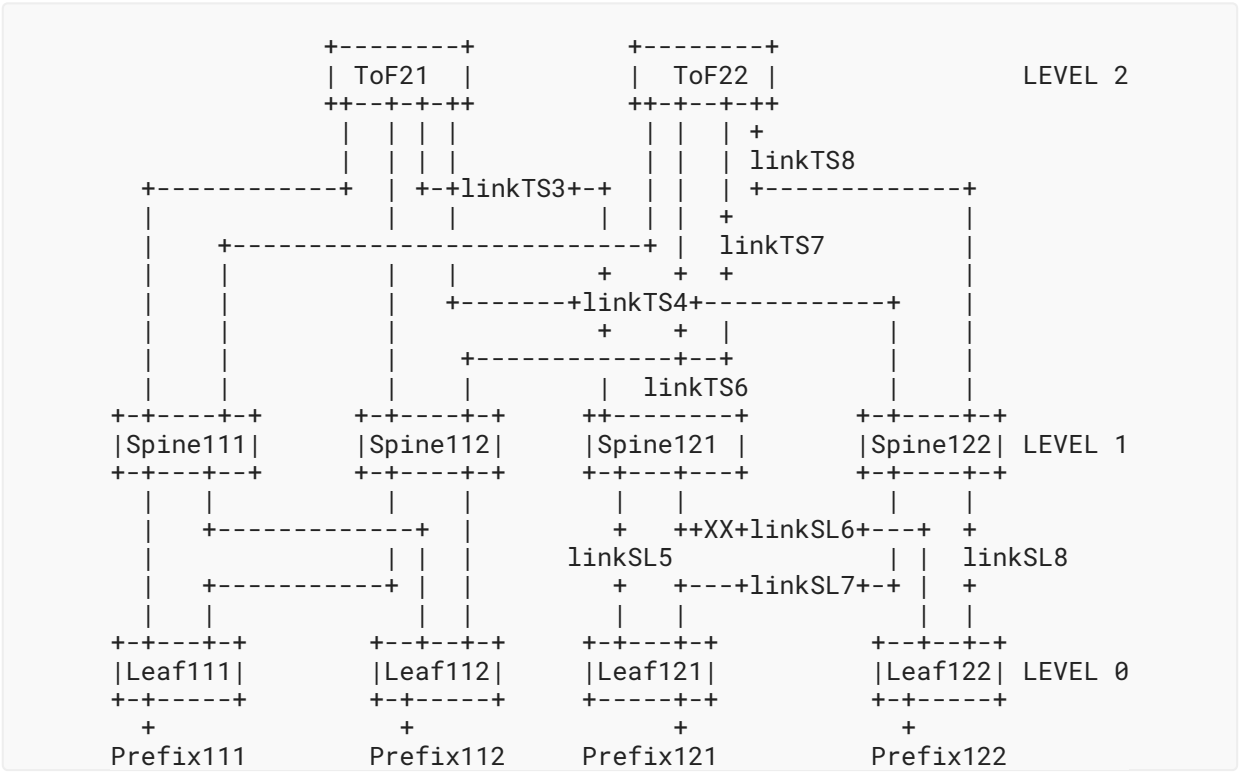

Figure 4: Suboptimal Routing Upon Link Failure Use Case

As shown in Figure 4, as the result of the south reflection, Spine121 and Spine 122 know each other via Leaf121 or Leaf 122 at level 1.

Without disaggregation mechanisms, the packet from leaf121 to prefix122 will probably go up through linkSL5 to linkTS3 when linkSL6 fails. Then, the packet will go down through linkTS4 to linkSL8 to Leaf122 or go up through linkSL5 to linkTS6, then go down through linkTS8 and linkSL8 to Leaf122 based on the pure default route. This is the case of suboptimal routing or bow tying.

With disaggregation mechanisms, Spine122 will detect the failure according to the reflected node S-TIE from Spine121 when linkSL6 fails. Based on the disaggregation algorithm provided by RIFT, Spine122 will explicitly advertise prefix122 in Disaggregated Prefix S-TIE PrefixTIEElement(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 it will go down through linkSL8 to Leaf122.

5.3. Black-Holing on Link Failures

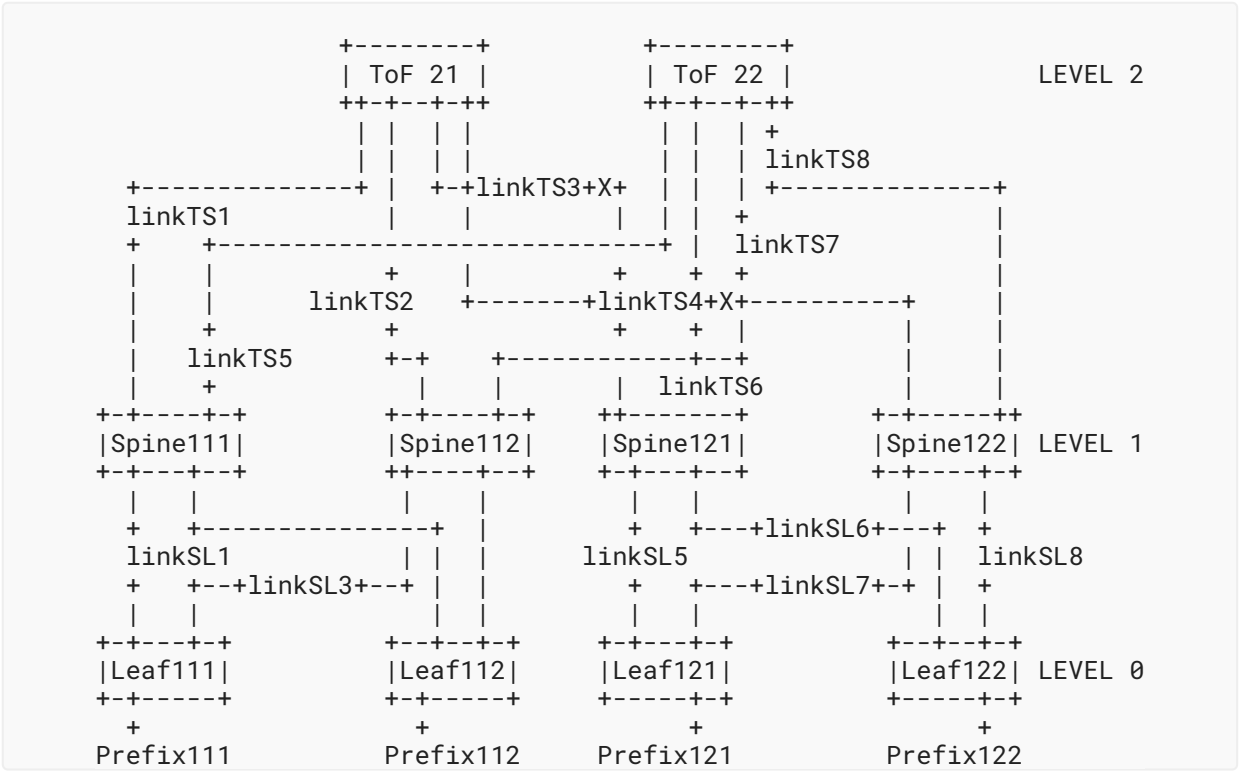

Figure 5: Black-Holing Upon Link Failure Use Case

This scenario illustrates a case where double link failure occurs and black-holing can happen.

Without disaggregation mechanisms, the packet from leaf111 to prefix122 would suffer 50% black-holing based on pure default route when linkTS3 and linkTS4 both fail. The packet is supposed to go up through linkSL1 to linkTS1 and then go down through linkTS3 or linkTS4 will be dropped. The packet is supposed to go up through linkSL3 to linkTS2, then go down through linkTS3 or linkTS4 will be dropped as well. This is the case of black-holing.

With disaggregation mechanisms, ToF22 will detect the failure according to the reflected node S-TIE of ToF21 from Spine111\Spine112 when linkTS3 and linkTS4 both fail. Based on the disaggregation algorithm provided by RIFT, 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.

5.4. Zero Touch Provisioning (ZTP)

RIFT is designed to require a very minimal configuration to simplify its operation and avoid human errors; based on that minimal information, ZTP auto configures the key operational parameters of all the RIFT nodes, including the System ID of the node that must be unique in the RIFT network and the level of the node in the Fat Tree, which determines which peers are northward "parents" and which are southward "children".

ZTP is always on, but its decisions can be overridden when a network administrator prefers to impose its own configuration. In that case, it is the responsibility of the administrator to ensure that the configured parameters are correct, i.e., ensure that the System ID of each node is unique and that the administratively set levels truly reflect the relative position of the nodes in the fabric. It is recommended to let ZTP configure the network, and when ZTP does not configure the network, it is recommended to configure the level of all the nodes to avoid an undesirable interaction between ZTP and the manual configuration.

ZTP requires that the administrator points out the ToF nodes to set the baseline from which the fabric topology is derived. The ToF nodes are configured with the TOP_OF_FABRIC flag, which are initial 'seeds' needed for other ZTP nodes to derive their level in the topology. ZTP computes the level of each node based on the Highest Available Level (HAL) of the potential parent closest to that baseline, which represents the superspine. In a fashion, RIFT can be seen as a distance-vector protocol that computes a set of feasible successors towards the superspine and autoconfigures the rest of the topology.

The autoconfiguration mechanism computes a global maximum of levels by diffusion. The derivation of the level of each node happens then based on LIEs received from its neighbors, whereas each node (with possible exceptions of configured leaves) 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 levels derived from nodes topologically below it and properly peer with nodes above.

The achieved equilibrium can be disturbed massively by all nodes with the highest level either leaving or entering the domain (with some finer distinctions not explained further). It is therefore recommended that each node is multihomed towards nodes with respective HAL offerings. Fortunately, this is the natural state of things for the topology variants considered in RIFT.

A RIFT node may also be configured to confine it to the leaf role with the LEAF_ONLY flag. A leaf node can also be configured to support leaf-2-leaf procedures with the LEAF_2_LEAF flag. In both cases, the node cannot be TOP_OF_FABRIC and its level cannot be configured. RIFT will fully determine the node's level after it is attached to the topology and ensure that the node is at the "bottom of the hierarchy" (southernmost).

5.5. Miscabling

5.5.1. Miscabling Examples

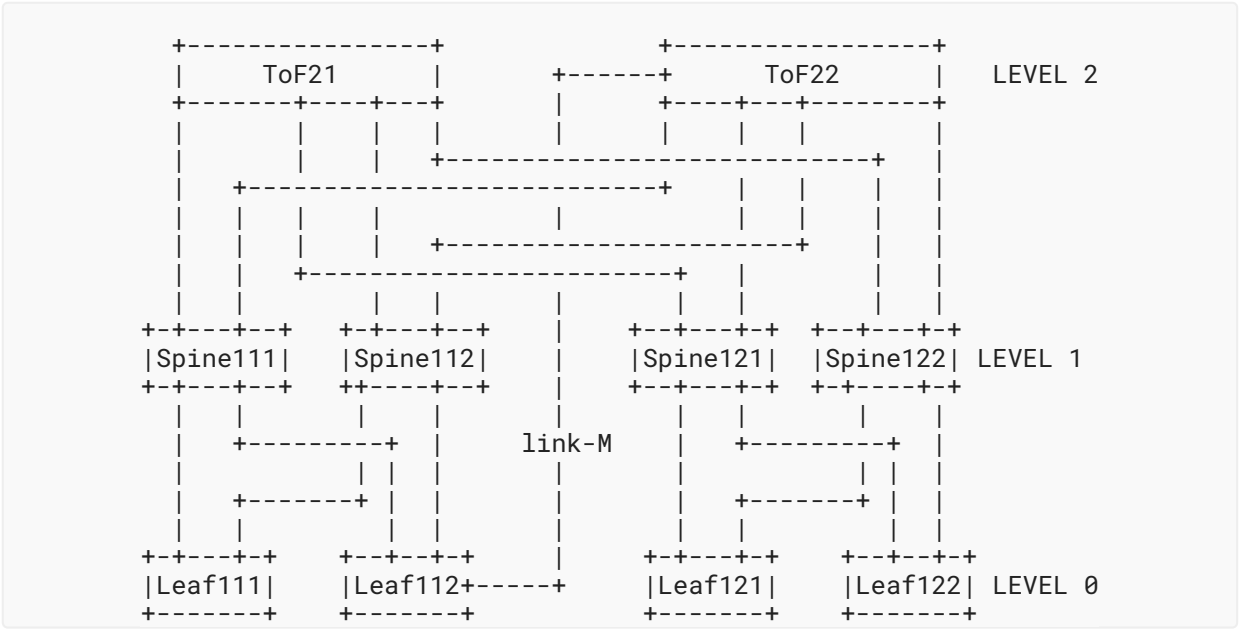

Figure 6: A Single-Plane Miscabling Example

Figure 6 shows a single-plane miscabling example. It's a perfect Fat Tree fabric except for link-M connecting Leaf112 to ToF22.

The RIFT control protocol can discover the physical links automatically and is able to detect cabling that violates Fat Tree topology constraints. It reacts accordingly to such miscabling attempts, preventing adjacencies between nodes from being formed and traffic from being forwarded on those miscabled links at a minimum. In such scenario, Leaf112 will use link-M to derive its level (unless it is leaf) and can report links to Spine111 and Spine112 as miscabled unless the implementations allow horizontal links.

Figure 7 shows a multi-plane miscabling example. Since Leaf112 and Spine121 belong to two different PoDs, the adjacency between Leaf112 and Spine121 cannot be formed. Link-W would be detected and prevented.

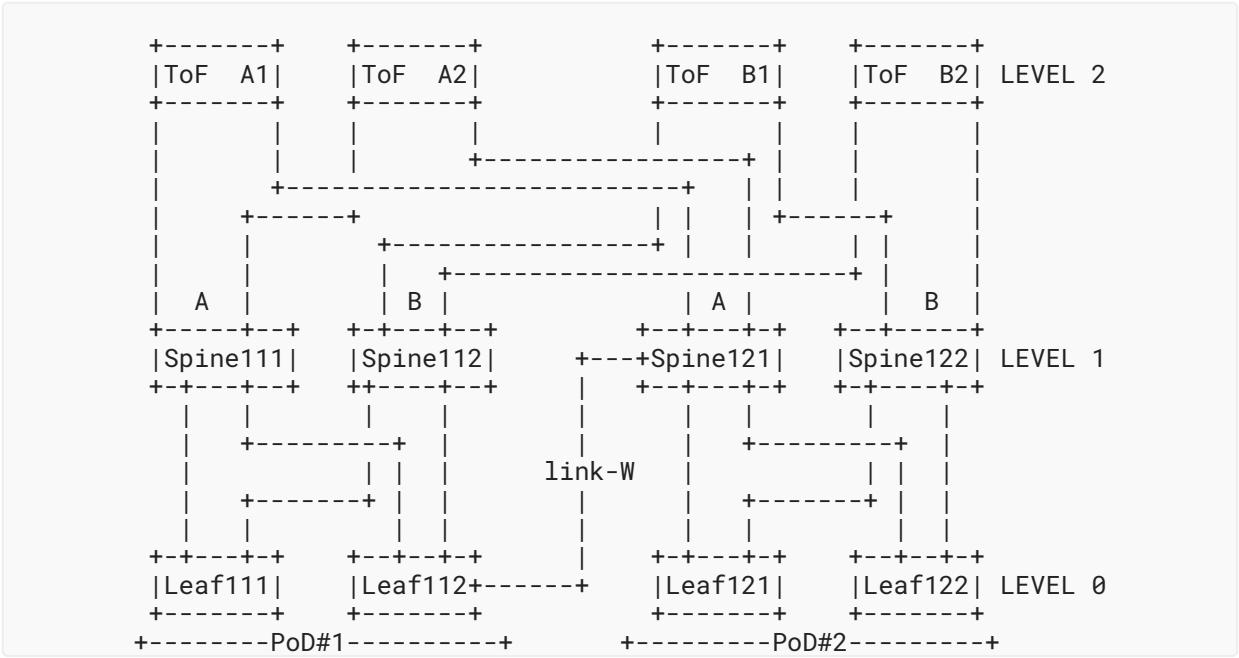

Figure 7: A Multiple Plane Miscabling Example

RIFT provides an optional level determination procedure in its ZTP mode. Nodes in the fabric without their level configured determine it automatically. However, this can have possible counter-intuitive consequences. One extreme failure scenario is depicted in Figure 8, and it shows that if all northbound links of Spine11 fail at the same time, Spine11 negotiates a lower level than Leaf11 and Leaf12.

To prevent such scenario where leaves are expected to act as switches, the LEAF_ONLY flag can be set for Leaf111 and Leaf112. Since level -1 is invalid, Spine11 would not derive a valid level from the topology in Figure 8. It will be isolated from the whole fabric, and it would be up to the leaves to declare the links towards such spine as miscabled.

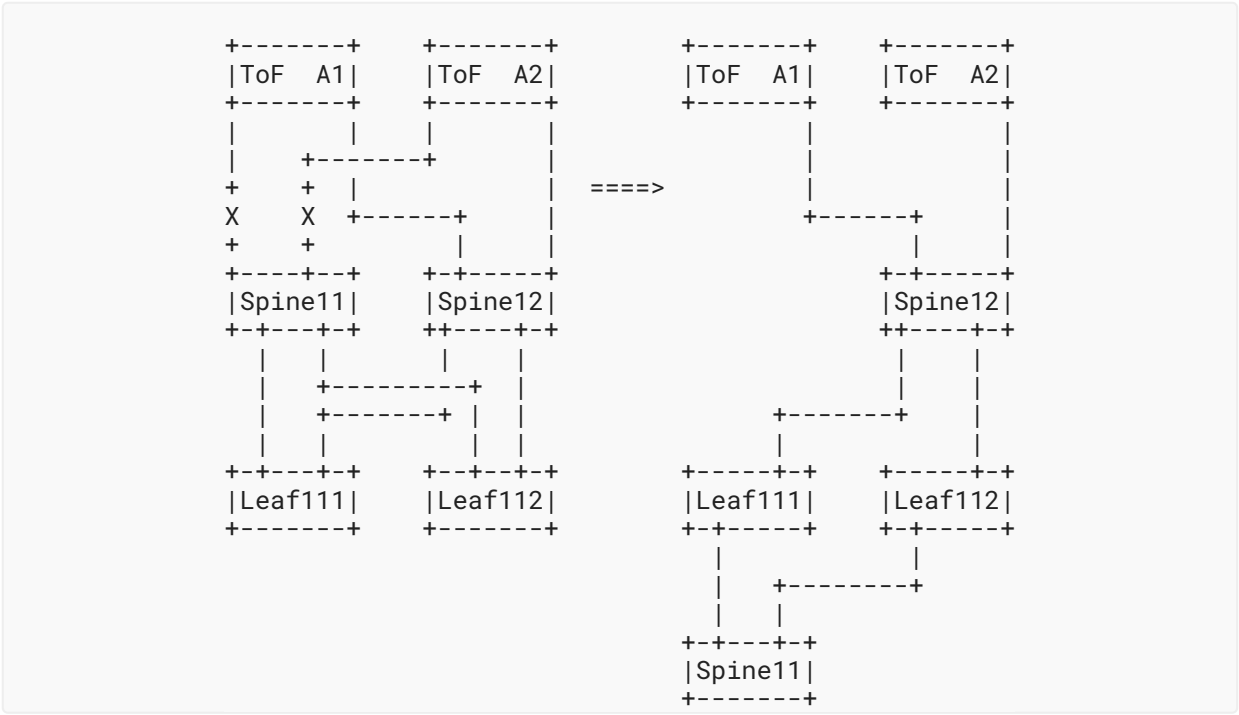

Figure 8: Fallen Spine

5.5.2. Miscabling Considerations

There are scenarios where operators may want to leverage ZTP and implement additional cabling constraints that go beyond the previously described topology violations. Enforcing cabling down to specific level, node, and port combinations might make it simpler for onsite staff to perform troubleshooting activities or replace optical transceivers and/or cabling as the physical layout will be consistent across the fabric. This is especially true for densely connected fabrics where it is difficult to physically manipulate those components. It is also easy to imagine other models, such as one where the strict port requirement is relaxed.

Figure 9 illustrates an example where the first port on Leaf1 must connect to the first port on Spine1, the second port on Leaf1 must connect to the first port on Spine2, and so on. Consider a case where (Leaf1, Port1) and (Leaf1, Port2) were reversed. RIFT would not consider this to be miscabled by default; however, an operator might want to.

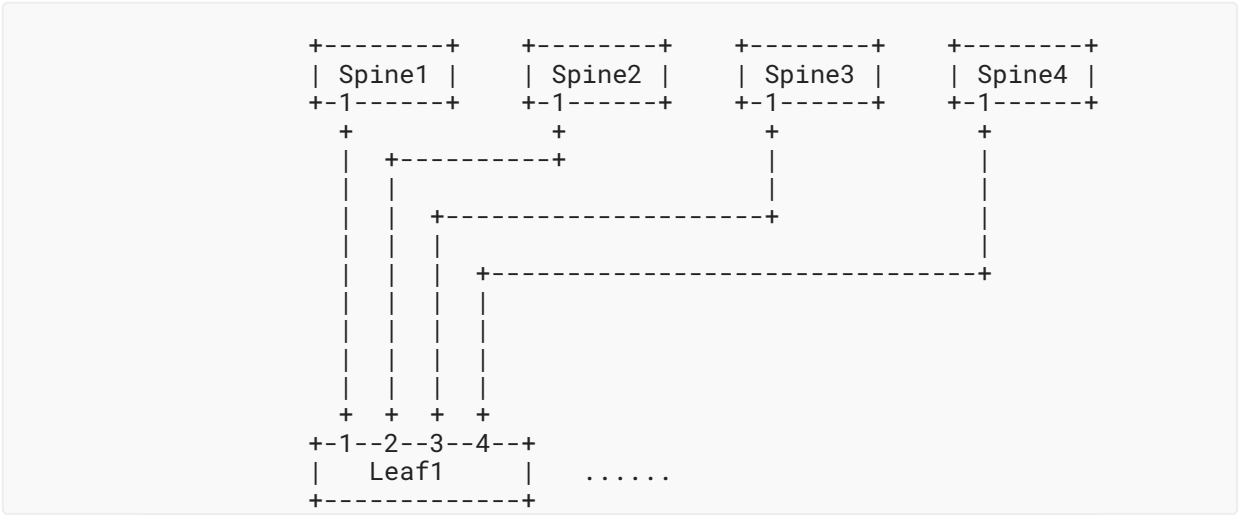

Figure 9: Additional Cabling Constraint Example

RIFT allows implementations to provide programmable plug-ins that can adjust ZTP operation or capture information during computation. While defining this is outside the scope of this document, such a mechanism could be used to extend the miscabling functionality.

For other protocols to achieve this, it would require additional operational overhead. Consider a fabric that is using unnumbered OSPF links; it is still very likely that a miscabled link will form an adjacency. Each attempt to move cables to the correct port may result in the need for additional troubleshooting as other links will become miscabled in the process. Without automation to explicitly tell the operator which ports need to be moved where, the process becomes manually intensive and error-prone very quickly. If the problem goes unnoticed, it will result in suboptimal performance in the fabric.

5.6. Multicast and Broadcast Implementations

RIFT supports both multicast and broadcast implementations. While a multicast implementation is preferred, there might cases where a broadcast implementation is optimal or even required. For example, operating systems on IoT devices and embedded devices may not have the required multicast support. Another example is containers, which do support multicast in some cases but tend to be very CPU-inefficient and difficult to tune.

5.7. Positive vs. Negative Disaggregation

Disaggregation is the procedure whereby RIFT [RFC9692] advertises a more specific route southwards as an exception to the aggregated fabric-default north. Disaggregation is useful when a prefix within the aggregation is reachable via some of the parents but not the others at the same level of the fabric. It is mandatory when the level is the ToF since a ToF node that cannot reach a prefix becomes a black hole for that prefix. The hard problem is to know which prefixes are reachable by whom.

In the general case, RIFT [RFC9692] solves that problem by interconnecting the ToF nodes so that the ToF nodes can exchange the full list of prefixes that exist in the fabric and figure out when a ToF node lacks reachability to some prefixes. This requires additional ports at the ToF, typically two ports per ToF node to form a ToF-spanning ring. RIFT [RFC9692] also defines the southbound reflection procedure that enables a parent to explore the direct connectivity of its peers, meaning their own parents and children; based on the advertisements received from the shared parents and children, it may enable the parent to infer the prefixes its peers can reach.

When a parent lacks reachability to a prefix, it may disaggregate the prefix negatively, i.e., advertise that this parent can be used to reach any prefix in the aggregation except that one. The Negative Disaggregation signaling is simple and functions transitively from ToF to Top-of-Pod (ToP) and then from ToP to Leaf. However, it is hard for a parent to figure out which prefix it needs to disaggregate because it does not know what it does not know; it results that the use of a spanning ring at the ToF is required to operate the Negative Disaggregation. Also, though it is only an implementation problem, the programming of the FIB is complex compared to normal routes and may incur recursions.

The more classical alternative is, for the parents that can reach a prefix that peers at the same level cannot, to advertise a more specific route to that prefix. This leverages the normal longest prefix match in the FIB and does not require a special implementation. As opposed to the Negative Disaggregation, the Positive Disaggregation is difficult and inefficient to operate transitively.

Transitivity is not needed by a grandchild if all its parents received the Positive Disaggregation, meaning that they shall all avoid the black hole; when that is the case, they collectively build a ceiling that protects the grandchild. Until then, a parent that received the Positive Disaggregation may believe that some peers are lacking the reachability and re-advertise too early or defer and maintain a black hole situation longer than necessary.

In a non-partitioned fabric, all the ToF nodes see one another through the reflection and can figure out if one is missing a child. In that case, it is possible to compute the prefixes that the peer cannot reach and disaggregate positively without a ToF-spanning ring. The ToF nodes can also ascertain that the ToP nodes are each connected to at least a ToF node that can still reach the prefix, meaning that the transitive operation is not required.

The bottom line is that in a fabric that is partitioned (e.g., using multiple planes) and/or where the ToP nodes are not guaranteed to always form a ceiling for their children, it is mandatory to use Negative Disaggregation. On the other hand, in a highly symmetrical and fully connected fabric (e.g., a canonical Clos Network), the Positive Disaggregation methods save the complexity and cost associated to the ToF-spanning ring.

Note that in the case of Positive Disaggregation, the first ToF nodes that announce a more-specific route attract all the traffic for that route and may suffer from a transient incast. A ToP node that defers injecting the longer prefix in the FIB, in order to receive more advertisements and spread the packets better, also keeps on sending a portion of the traffic to the black hole in the

meantime. In the case of Negative Disaggregation, the last ToF nodes that inject the route may also incur an incast issue; this problem would occur if a prefix that becomes totally unreachable is disaggregated.

5.8. Mobile Edge and Anycast

When a physical or a virtual node changes its point of attachment in the fabric from a previous-leaf to a next-leaf, new routes must be installed that supersede the old ones. Since the flooding flows northwards, the nodes (if any) between the previous-leaf and the common parent are not immediately aware that the path via the previous-leaf is obsolete and a stale route may exist for a while. The common parent needs to select the freshest route advertisement in order to install the correct route via the next-leaf. This requires that the fabric determines the sequence of the movements of the mobile node.

On the one hand, a classical sequence counter provides a total order for a while, but it will eventually wrap. On the other hand, a timestamp provides a permanent order, but it may miss a movement that happens too quickly vs. the granularity of the timing information. It is not envisioned that an average fabric supports the Precision Time Protocol [IEEEstd1588] in the short term nor that the precision available with the Network Time Protocol [RFC5905] (in the order of 100 to 200 ms) may not be necessarily enough to cover, e.g., the fast mobility of a Virtual Machine (VM).

Section 6.8.4 ("Mobility") of [RFC9692] specifies a hybrid method that combines a sequence counter from the mobile node and a timestamp from the network taken at the leaf when the route is injected. If the timestamps of the concurrent advertisements are comparable (i.e., more distant than the precision of the timing protocol), then the timestamp alone is used to determine the relative freshness of the routes. Otherwise, the sequence counter from the mobile node is used if it is available. One caveat is that the sequence counter must not wrap within the precision of the timing protocol. Another is that the mobile node may not even provide a sequence counter; in which case, the mobility itself must be slower than the precision of the timing.

Mobility must not be confused with anycast. In both cases, the same address is injected in RIFT at different leaves. In the case of mobility, only the freshest route must be conserved since the mobile node changes its point of attachment for a leaf to the next. In the case of anycast, the node may either be multihomed (attached to multiple leaves in parallel) or reachable beyond the fabric via multiple routes that are redistributed to different leaves. Either way, the multiple routes are equally valid and should be conserved in the case of anycast. Without further information from the redistributed routing protocol, it is impossible to sort out a movement from a redistribution that happens asynchronously on different leaves. RIFT [RFC9692] expects that anycast addresses are advertised within the timing precision, which is typically the case with a low-precision timing and a multihomed node. Beyond that time interval, RIFT interprets the lag as a mobility and only the freshest route is retained.

When using IPv6 [RFC8200], RIFT suggests leveraging 6LoWPAN ND [RFC8505] as the IPv6 ND interaction between the mobile node and the leaf. This not only provides a sequence counter but also a lifetime and a security token that may be used to protect the ownership of an address [RFC8928]. When using 6LoWPAN ND [RFC8505], the parallel registration of an anycast address

to multiple leaves is done with the same sequence counter, whereas the sequence counter is incremented when the point of attachment changes. This way, it is possible to differentiate a mobile node from a multihomed node, even when the mobility happens within the timing precision. It is also possible for a mobile node to be multihomed as well, e.g., to change only one of its points of attachment.

5.9. IPv4 over IPv6

RIFT allows advertising IPv4 prefixes over an IPv6 RIFT network. An IPv6 Address Family (AF) configures via the usual ND mechanisms and then V4 can use V6 next-hops analogous to [RFC8950]. It is expected that the whole fabric supports the same type of forwarding of AFs on all the links. RIFT provides an indication whether a node is capable of V4-forwarding and implementations are possible where different routing tables are computed per AF as long as the computation remains loop-free.

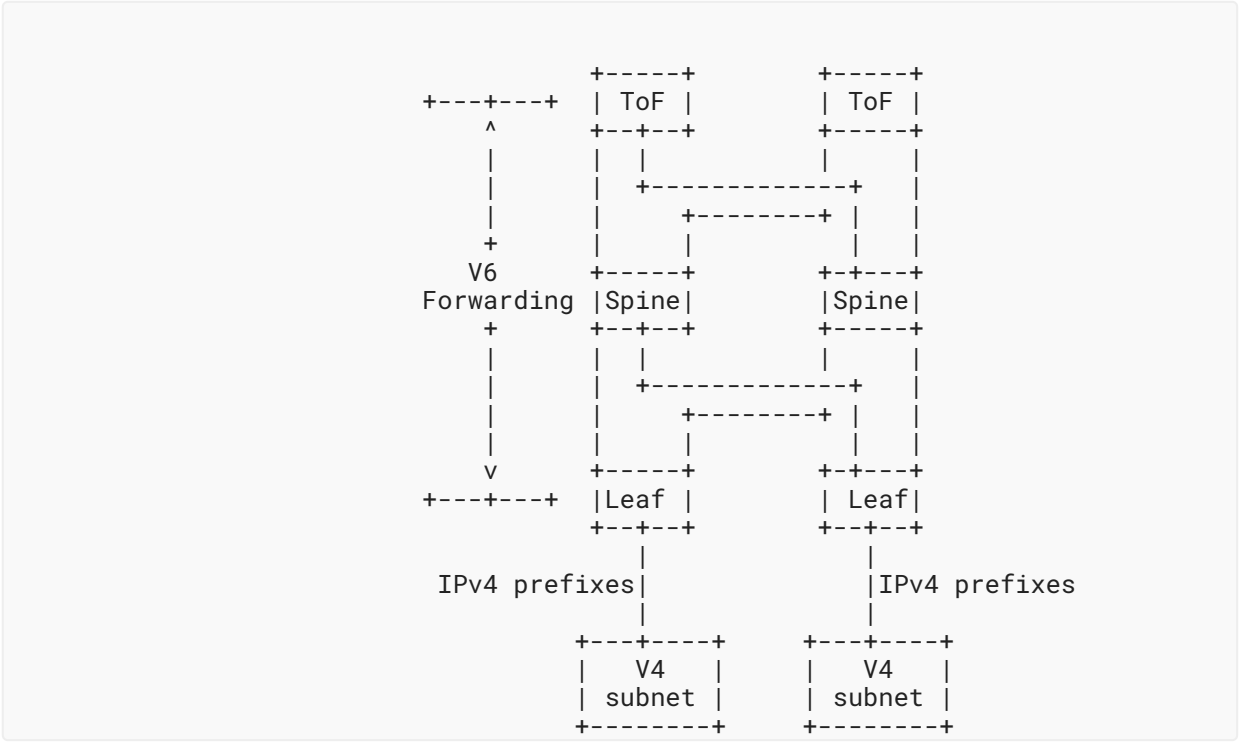

Figure 10: IPv4 over IPv6

5.10. In-Band Reachability of Nodes

RIFT doesn't precondition that nodes of the fabric have reachable addresses, but the operational reasons to reach the internal nodes may exist. Figure 11 shows an example that the network management station (NMS) attaches to Leaf1.

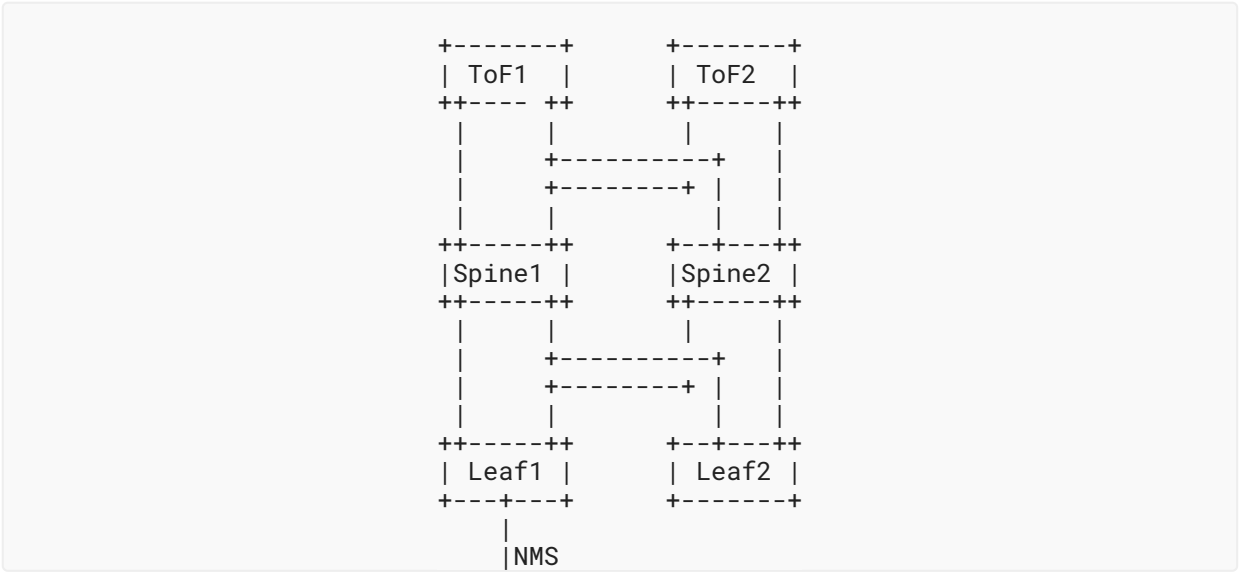

Figure 11: In-Band Reachability of Nodes

If the NMS wants to access Leaf2, it simply works because the loopback address of Leaf2 is flooded in its Prefix North TIE.

If the NMS wants to access Spine2, it also works because a spine node always advertises its loopback address in the Prefix North TIE. The NMS may reach Spine2 from Leaf1-Spine2 or Leaf1-Spine1-ToF1/ToF2-Spine2.

If the NMS wants to access ToF2, ToF2's loopback address needs to be injected into its Prefix South TIE. This TIE must be seen by all nodes at the level below -- the spine nodes in Figure 11 -- that must form a ceiling for all the traffic coming from below (south). Otherwise, the traffic from the NMS may follow the default route to the wrong ToF Node, e.g., ToF1.

In the case of failure between ToF2 and spine nodes, ToF2's loopback address must be disaggregated recursively all the way to the leaves. In a partitioned ToF, even with recursive disaggregation, a ToF node is only reachable within its plane.

A possible alternative to recursive disaggregation is to use a ring that interconnects the ToF nodes to transmit packets between them for their loopback addresses only. The idea is that this is mostly control traffic and should not alter the load-balancing properties of the fabric.

5.11. Dual-Homing Servers

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

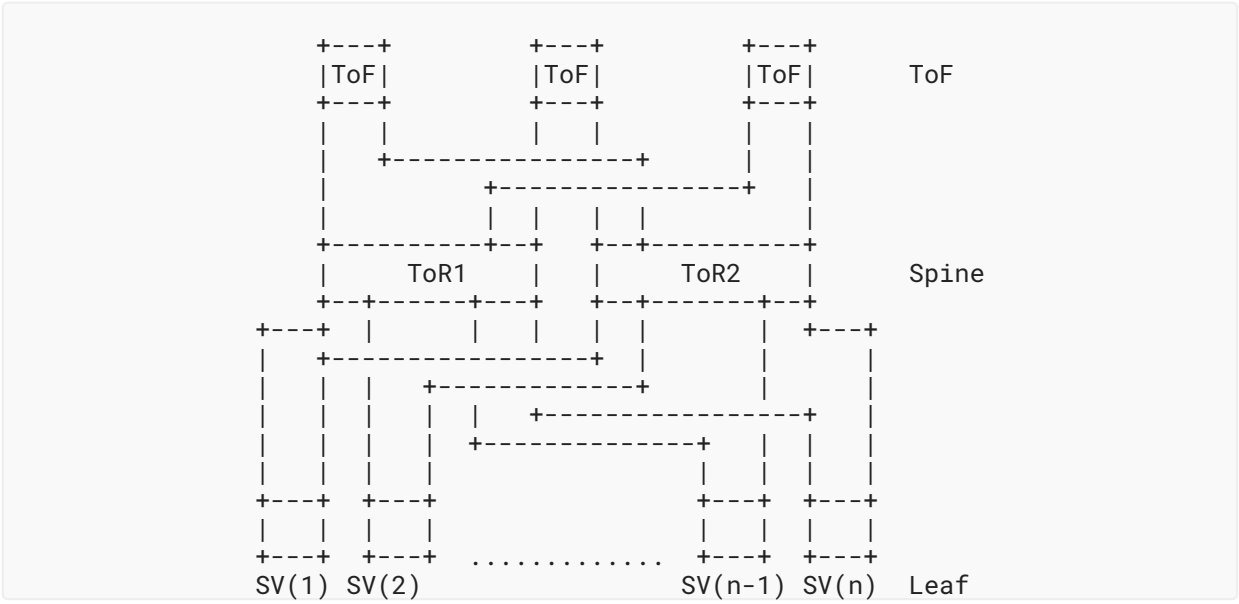

Figure 12: Dual-Homing Servers

Sometimes people may prefer to disaggregate from ToR nodes to servers from startup, i.e., the servers have multiple routes in the FIB from startup other than default routes to avoid breakages at the rack level. Full disaggregation of the fabric could be achieved by configuration supported by RIFT.

5.12. Fabric with a Controller

There are many different ways to deploy the controller. One possibility is attaching a controller to the RIFT domain from ToF and another possibility is attaching a controller from the leaf.

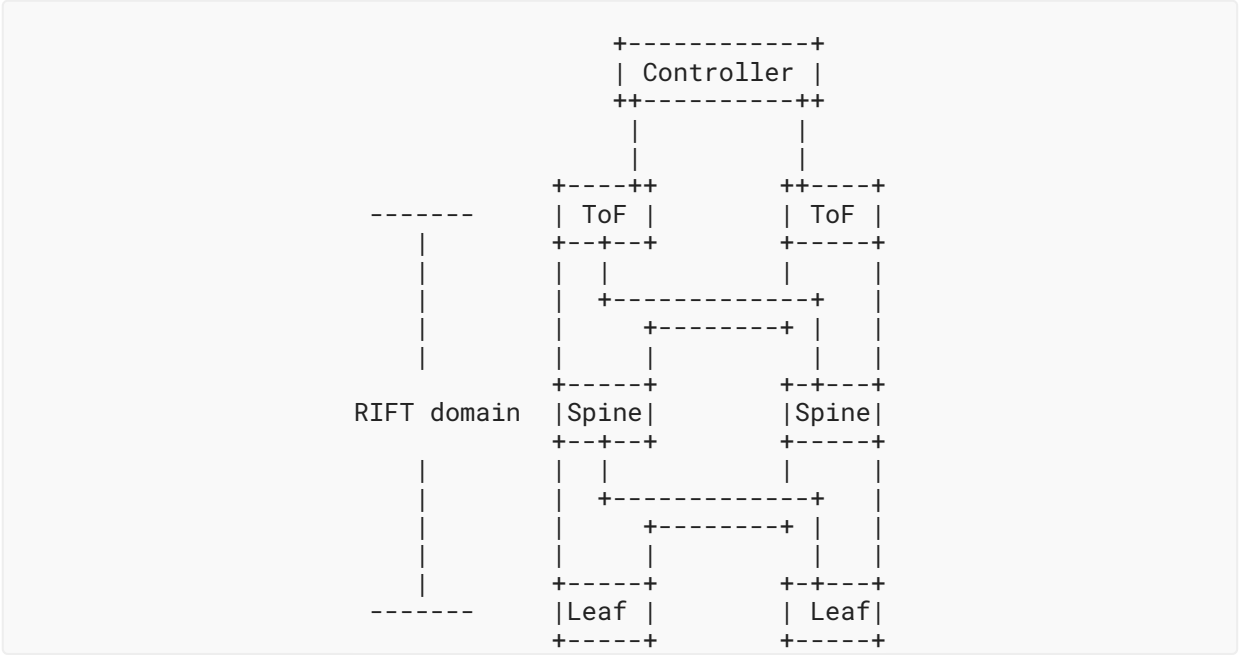

Figure 13: Fabric with a Controller

5.12.1. Controller Attached to ToFs

If a controller is attaching to the RIFT domain from ToF, it usually uses dual-homing connections. The loopback prefix of the controller should be advertised down by the ToF and spine to the leaves. If the controller loses the link to ToF, make sure the ToF withdraws the prefix of the controller.

5.12.2. Controller Attached to Leaf

If the controller is attaching from a leaf to the fabric, no special provisions are needed.

5.13. Internet Connectivity Within Underlay

If global addressing is running without overlay, an external default route needs to be advertised through the RIFT fabric to achieve internet connectivity. For the purpose of forwarding of the entire RIFT fabric, an internal fabric prefix needs to be advertised in the Prefix South TIE by ToF and spine nodes.

5.13.1. Internet Default on the Leaf

In the case that the internet gateway is a leaf, the leaf node as the internet gateway needs to advertise a default route in its Prefix North TIE.

5.13.2. Internet Default on the ToFs

In the case that the internet gateway is a ToF, the ToF and spine nodes need to advertise a default route in the Prefix South TIE.

5.14. Subnet Mismatch and Address Families

Figure 14: Subnet Mismatch

LIEs are exchanged over all links running RIFT to perform Link (Neighbor) Discovery. A node must NOT originate LIEs on an AF if it does not process received LIEs on that family. LIEs on the same link are considered part of the same negotiation independent from the AF they arrive on. An implementation must be ready to accept TIEs on all addresses it used as the source of LIE frames.

As shown in Figure 14, an adjacency of nodes A and B may form without further checks, but the forwarding between nodes A and B may fail because subnet X mismatches with subnet Y.

To prevent this, a RIFT implementation should check for subnet mismatch in a way that is similar to how IS-IS does. This can lead to scenarios where an adjacency, despite the exchange of LIEs in both AFs, may end up having an adjacency in a single AF only. This is especially a consideration in scenarios relating to Section 5.9.

5.15. Anycast Considerations

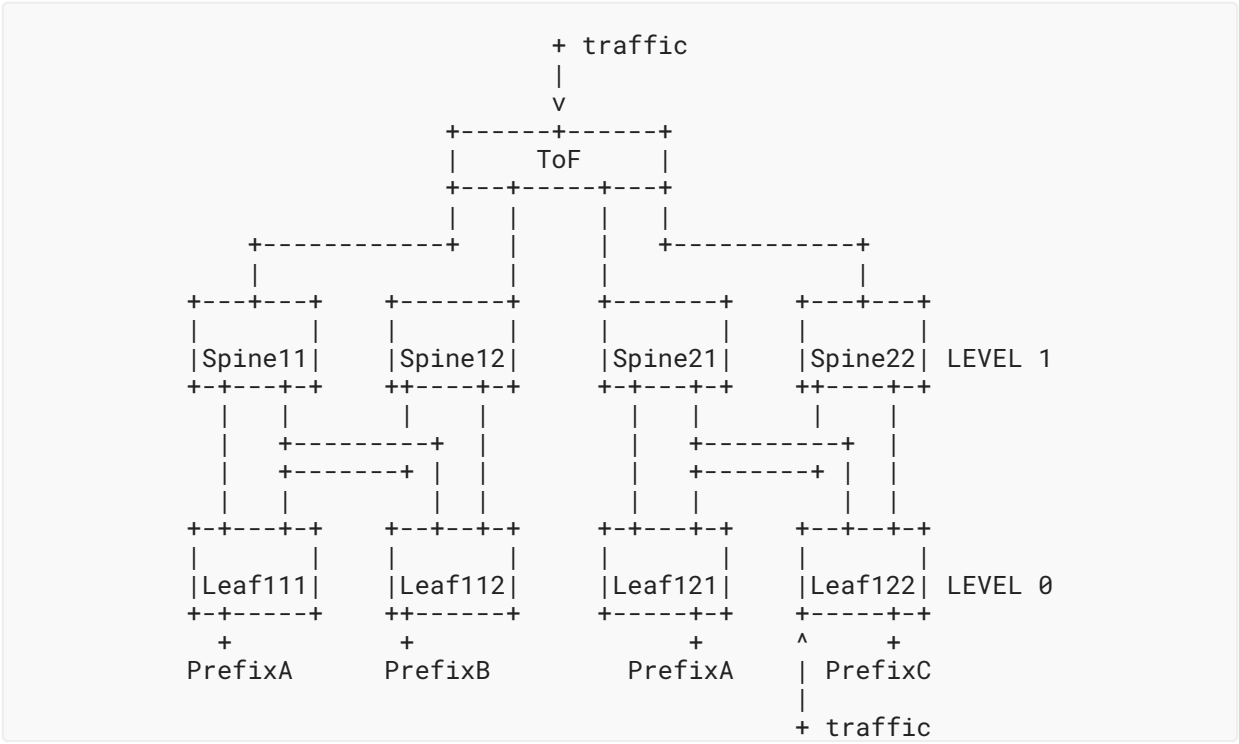

Figure 15: Anycast

If the traffic comes from ToF to Leaf111 or Leaf121, which has anycast prefix PrefixA, RIFT can deal with this case well. However, if the traffic comes from Leaf122, it arrives to Spine21 or Spine22 at LEVEL 1. Additionally, Spine21 or Spine22 doesn't know another PrefixA attaching Leaf111, so it will always get to Leaf121 and never Leaf111. If the intention is that the traffic should be offloaded to Leaf111, then use the policy-guided prefixes defined in RIFT [RFC9692].

5.16. IoT Applicability

The design of RIFT inherits the anisotropic design of a default route upwards (northwards) from RPL [RFC6550]. It also inherits the capability to inject external host routes at the Leaf level using Wireless ND (WiND) [RFC8505] [RFC8928] between a RIFT-agnostic host and a RIFT router. Both the RPL and the RIFT protocols are meant for a large scale, and WiND enables device mobility at the edge the same way in both cases.

The main difference between RIFT and RPL is that there's a single root with RPL, whereas RIFT has many ToF nodes. This adds huge capabilities for leaf-2-leaf ECMP paths but additional complexity with the need to disaggregate. Also, RIFT uses link-state flooding northwards and is not designed for low-power operation.

Still, nothing prevents that the IP devices connected at the Leaf are IoT devices, which typically expose their address using WiND -- this is an upgrade from 6LoWPAN ND [RFC6775].

A network that serves high speed / high power IoT devices should typically provide deterministic capabilities for applications such as high speed control loops or movement detection. The Fat Tree is highly reliable and, in normal conditions, provides an equivalent multipath operation; however, the ECMP doesn't provide hard guarantees for either delivery or latency. As long as the fabric is non-blocking, the result is the same, but there can be load unbalances resulting in incast and possibly congestion loss that will prevent the delivery within bounded latency.

This could be alleviated with Packet Replication, Elimination, and Ordering Functions (PREOF) [RFC8655] leaf-2-leaf, but PREOF is hard to provide at the scale of all flows and the replication may increase the probability of the overload that it attempts to solve.

Note that the load balancing is not RIFT's problem, but it is key to serve IoT adequately.

5.17. Key Management

As outlined in Section 9 ("Security Considerations") of [RFC9692], either a private shared key or a public/private key pair is used to authenticate the adjacency. Both the key distribution and key synchronization methods are out of scope for this document. Both nodes in the adjacency must share the same keys, key type, and algorithm for a given key ID. Mismatched keys will not interoperate as their security envelopes will be unverifiable.

Key rollover while the adjacency is active may be supported. The specific mechanism is well documented in [RFC6518]. As outlined in 9.9 ("Host Implementations") of [RFC9692], hosts as well as VMs acting as RIFT devices are possible. Key Management Protocols (KMPs), such as Key Value (KV) for key rollover in the fabric, use a symmetric key that can be changed easily when compromised; in which case, the symmetric key of a host is more likely to be compromised than an in-fabric networking node.

5.18. TTL/Hop Limit of 1 vs. 255 on LIEs/TIEs

The use of a packet's Time to Live (TTL) (IPv4) or Hop Limit (IPv6) to verify whether the packet was originated by an adjacent node on a connected link has been used in RIFT. RIFT explicitly requires the use of a TTL/HL value of 1 or 255 when sending/receiving LIEs and TIEs so that implementers have a choice between the two.

TTL=1 or HL=1 protects against the information disseminating more than 1 hop in the fabric and should be the default unless configured otherwise. TTL=255 or HL=255 can lead RIFT TIE packet propagation to more than one hop (the multicast address is already in local subnetwork range) in case of implementation problems but does protect against a remote attack as well, and the receiving remote router will ignore such TIE packet unless the remote router is exactly 254 hops away and accepts only TTL=1 or HL=1. [RFC5082] defines a Generalized TTL Security Mechanism (GTSM). The GTSM is applicable to LIE/TIE implementations that use a TTL or HL of 255. It provides a defense from infrastructure attacks based on forged protocol packets from outside the fabric.

6. Security Considerations

This document presents applicability of RIFT. As such, it does not introduce any security considerations. However, there are a number of security concerns in [RFC9692].

7. IANA Considerations

This document has no IANA actions.

8. References

8.1. Normative References

[ISO10589-Second-Edition] ISO/IEC, "Information technology - Telecommunications and information exchange between systems - 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)", ISO/IEC 10589:2002, November 2002, <https://www.iso.org/standard/30932.html>.

[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>.

[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, <https://www.rfc-editor.org/info/rfc5082>.

[RFC5340] Coltun, R., Ferguson, D., Moy, J., and A. Lindem, "OSPF for IPv6", RFC 5340, DOI 10.17487/RFC5340, July 2008, <https://www.rfc-editor.org/info/rfc5340>.

[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>.

[RFC6518] Lebovitz, G. and M. Bhatia, "Keying and Authentication for Routing Protocols (KARP) Design Guidelines", RFC 6518, DOI 10.17487/RFC6518, February 2012, <https://www.rfc-editor.org/info/rfc6518>.

[RFC6550] Winter, T., Ed., Thubert, P., Ed., Brandt, A., Hui, J., Kelsey, R., Levis, P., Pister, K., Struik, R., Vasseur, JP., and R. Alexander, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks", RFC 6550, DOI 10.17487/RFC6550, March 2012, <https://www.rfc-editor.org/info/rfc6550>.

[RFC6775] Shelby, Z., Ed., Chakrabarti, S., Nordmark, E., and C. Bormann, "Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)", RFC 6775, DOI 10.17487/RFC6775, November 2012, <https://www.rfc-editor.org/info/rfc6775>.

[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>.

[RFC8655] Finn, N., Thubert, P., Varga, B., and J. Farkas, "Deterministic Networking Architecture", RFC 8655, DOI 10.17487/RFC8655, October 2019, <https://www.rfc-editor.org/info/rfc8655>.

[RFC8950] Litkowski, S., Agrawal, S., Ananthamurthy, K., and K. Patel, "Advertising IPv4 Network Layer Reachability Information (NLRI) with an IPv6 Next Hop", RFC 8950, DOI 10.17487/RFC8950, November 2020, <https://www.rfc-editor.org/info/rfc8950>.

[RFC9692] Przygienda, T., Ed., Head, J., Ed., Sharma, A., Thubert, P., Rijsman, B., and D. Afanasiev, "RIFT: Routing in Fat Trees", RFC 9692, DOI 10.17487/RFC9692, December 2024, <https://www.rfc-editor.org/info/rfc9692>.

[TR-384] Broadband Forum Technical Report, "TR-384: Cloud Central Office Reference Architectural Framework", TR-384, Issue 1, January 2018, <https://www.broadband-forum.org/pdfs/tr-384-1-0-0.pdf>.

8.2. Informative References

[CLOS] Yuan, X., "On Nonblocking Folded-Clos Networks in Computer Communication Environments", 2011 IEEE International Parallel & Distributed Processing Symposium, DOI 10.1109/IPDPS.2011.27, May 2011, <https://ieeexplore.ieee.org/document/6012836>.

[FATTREE] Leiserson, C. E., "Fat-Trees: Universal Networks for Hardware-Efficient Supercomputing", IEEE Transactions on Computers, vol. C-34, no. 10, pp. 892-901, DOI 10.1109/TC.1985.6312192, October 1985, <https://ieeexplore.ieee.org/document/6312192>.

[IEEEstd1588] IEEE, "IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems", IEEE Std 1588-2019, DOI 10.1109/IEEESTD.2020.9120376, June 2020, <https://ieeexplore.ieee.org/document/9120376>.

[PNNI] The ATM Forum Technical Committee, "Private Network-Network Interface - Specification Version 1.1 - (PNNI 1.1)", af-pnni-0055.001, April 2002, <https://www.broadband-forum.org/download/af-pnni-0055.001.pdf>.

[RFC3626] Clausen, T., Ed. and P. Jacquet, Ed., "Optimized Link State Routing Protocol (OLSR)", RFC 3626, DOI 10.17487/RFC3626, October 2003, <https://www.rfc-editor.org/info/rfc3626>.

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

[RFC5905] Mills, D., Martin, J., Ed., Burbank, J., and W. Kasch, "Network Time Protocol Version 4: Protocol and Algorithms Specification", RFC 5905, DOI 10.17487/RFC5905, June 2010, <https://www.rfc-editor.org/info/rfc5905>.

[RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <https://www.rfc-editor.org/info/rfc8200>.

[RFC8505] Thubert, P., Ed., Nordmark, E., Chakrabarti, S., and C. Perkins, "Registration Extensions for IPv6 over Low-Power Wireless Personal Area Network (6LoWPAN) Neighbor Discovery", RFC 8505, DOI 10.17487/RFC8505, November 2018, <https://www.rfc-editor.org/info/rfc8505>.

[RFC8928] Thubert, P., Ed., Sarikaya, B., Sethi, M., and R. Struik, "Address-Protected Neighbor Discovery for Low-Power and Lossy Networks", RFC 8928, DOI 10.17487/RFC8928, November 2020, <https://www.rfc-editor.org/info/rfc8928>.

Acknowledgments

The authors would like to thank Jaroslaw Kowalczyk, Alvaro Retana, Jim Guichard, and Jeffrey Zhang for providing invaluable concepts and content for this document.

Contributors

The following people contributed substantially to the content of this document and should be considered coauthors:

Jordan Head
Juniper Networks
Email: jhead@juniper.net

Tom Verhaeg
Juniper Networks
Email: tverhaeg@juniper.net

Authors' Addresses

Yuehua Wei (EDITOR)
ZTE Corporation
No.50, Software Avenue
Nanjing
210012
China
Email: wei.yuehua@zte.com.cn

Zheng (Sandy) Zhang
ZTE Corporation
No.50, Software Avenue
Nanjing
210012
China
Email: zhang.zheng@zte.com.cn

Dmitry Afanasiev
Yandex
Email: fl0w@yandex-team.ru

Pascal Thubert
Individual
France
Email: pascal.thubert@gmail.com

Tony Przygienda
Juniper Networks
1194 N. Mathilda Ave
Sunnyvale, CA 94089
United States of America
Email: prz@juniper.net