

RIFT
Internet-Draft
Intended status: Standards Track
Expires: January 14, 2021

Z. Zhang
Juniper Networks
P. Thubert
Cisco
July 13, 2020

Multicast Routing In Fat Trees
draft-zzhang-rift-multicast-01

Abstract

This document specifies multicast procedures with RIFT. Multicast in RIFT is similar to Bidirectional Protocol Independent Multicast (PIM-Bidir), with the Rendezvous Point Link (RP-Link) simulated by a spanning tree of some Top of Fabric (TOF) nodes and sub-TOF nodes.

Requirements Language

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

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

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

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

This Internet-Draft will expire on January 14, 2021.

Copyright Notice

Copyright (c) 2020 IETF Trust and the persons identified as the document authors. All rights reserved.

Internet-Draft

mrift

July 2020

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

Table of Contents

1.	Introduction	2
2.	Specifications	4
2.1.	Multicast Capability	4
2.2.	Optional Per-neighbor Flooding Scope	5
2.3.	Multicast TIE	5
2.4.	Building Spanning Tree among TOFs and sub-TOFs	6
3.	Security Considerations	7
4.	Acknowledgements	7
5.	References	7
5.1.	Normative References	7
5.2.	Informative References	8
	Authors' Addresses	8

[1.](#) Introduction

Because of the simple north-south regular topology in Fat Tree networks, the PIM-Bidir [[RFC5015](#)] solution is extended for multicast in RIFT (referred to as MRIFT in this document). The following is a summary of the changes and adaptations compared to PIM-Bidir.

With PIM-Bidir, PIM joins are sent towards a Rendezvous Point Address (RPA), which could be an address not belonging to any router. The RPA does belong to a RP Link (RPL), which could be attached to a single router or multiple routers (e.g. RPL is a LAN). With MRIFT, there is no concept of RPA any more (joins are simply sent northbound). The joins are terminated on some sub-TOF nodes and the RPL is simulated by a spanning tree among some TOF and sub-TOF nodes.

Instead of (*,G) trees in PIM-Bidir, MRIFT uses (*,G-Prefix) trees, where the G-Prefix could be *, G, or anything in between (e.g., 225.1.1.0/24). For light flows, they could just follow the (*,*)

tree. For heavy flows, individual (*,G) trees could be built. For medium flows, some (*,G-prefix) trees could be shared. All the First Hop Routers (FHRs, connecting to sources) and the Last Hop Routers (LHRs, connecting to receivers) of a particular (*,G) flow must agree on whether a (*,*) or (*,G) or (*,G-prefix) tree is used for the flow

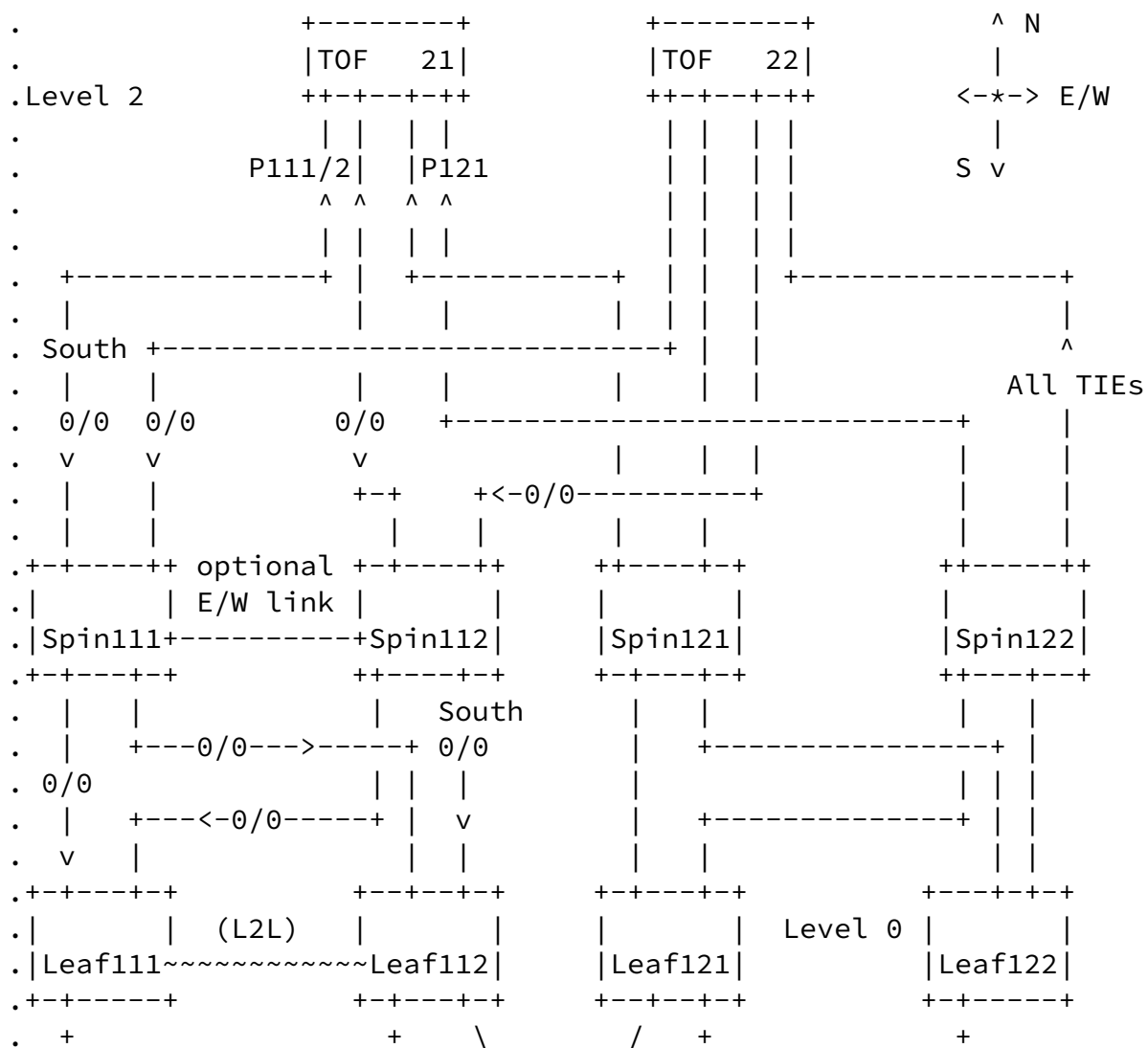
so that they all join the same tree. This is done via out of band control outside the scope of this document.

Because of the rich connections in Fat Trees, a router has to choose one of its many north neighbors to send join to. This is done through hashing. The hashing algorithm should lead to several but not too many routers choosing the same north neighbor, so that fewer routers are involved in multicast traffic forwarding, yet none of those routers are overburdened by replicating to too many downstream neighbors.

Instead of PIM messages, RIFT's own TIEs are used, similar to the concept in [[draft-zzhang-pim-pds](#)]. This introduces the concept of neighbor-scoped flooding - a multicast TIE is sent only to a chosen upstream north neighbor that consumes it and then regenerates a new TIE for the next upstream.

When a join reaches a sub-TOF node, the normal join process stops. This forms a sub-tree rooted at this sub-TOF node. Multiple sub-trees of the same tree may be joined by a single TOF node, or they may have to be connected by a spanning tree serving as the RPL. For example, in the following topology, in normal situations the two sub-tree roots for the two pods, say Spine111 and Spine121, may be joined by TOF21, but if the TOF21-Spine121 link is down, then TOF22 may be used, and if the TOF22-Spine111 link is also down, then Spine111 and Spine121 will have to be joined via Spine111-TOF21-Spine112-TOF22-Spine121.

July 2020



```

. Prefix111 Prefix112 \ / Prefix121 Prefix122
.
. multi-homed
. Prefix
.+----- Pod 1 -----+ +----- Pod 2 -----+

```

[2. Specifications](#)

[2.1. Multicast Capability](#)

A new optional field is added to the NodeCapabilities to indicate that the node is enabled for multicast:

```

struct NodeCapabilities {
    ...
    4: optional bool          multicast_enabled;
}

```

[2.2. Optional Per-neighbor Flooding Scope](#)

This document introduces an optional per-neighbor flooding scope for TIEs:

```

struct TIEHeader {
    ...
    13: optional common.SystemIDType flooding_scope_neighbor;
}

```

When a node originates a TIE with a per-neighbor flooding scope, it is sent to the specified neighbor only. When a node receives a TIE with per-neighbor flooding scope, it is accepted only if the node is the specified neighbor, and it is not reflooded any further.

[2.3. Multicast TIE](#)

Currently the multicast TIEs are only N-TIEs with per-neighbor flooding scope except on TOFs and sub-TOFs. If a multicast TIE is received from a node south of sub-TOFs without the per-neighbor flooding scope specified, it MUST be discarded.

```
/** TIE for multicast */
struct IPMulticastTIEElement {
    /** Multicast TIEs are for (*, group-prefix) joins.
        The '*' is not encoded in the TIE. */
    1: required common.IPPrefixType      group_prefix;

    /** fields used by TOFs and sub-TOFs to build spanning tree RPL */
    2: optional common.SystemIDType      chosen_or_highest_parent;
    3: optional list<common.SystemIDType> sub_tof_children;
}

/** Type of TIE.
    ...
*/
enum TIETypeType {
```

```

    ...
    TIETypeIPMulticast
    TIETypeMaxValue
}
= 11,
= 12,

/** Single element in a TIE.
    ...
*/
union TIEElement {
    ...
    /** IP multicast elements. */
    10: optional IPMulticastTIEElement ip_multicast;
}

```

[2.4.](#) Building Spanning Tree among TOFs and sub-TOFs

Note: this is still subject to further discussion/change. It may be replaced by another scheme upon further discussions.

If a sub-TOF node is the root of a sub-tree for a (*, G-prefix) tree, it hashes to a TOF neighbor as its parent for the tree, and originates a corresponding multicast N-TIE without the per-neighbor flooding scope - flooded to all its north TOF neighbors. The `chosen_or_highest_parent` field is set to the chosen TOF neighbor.

A receiving TOF node originates a corresponding S-TIE without the per-neighbor flooding scope. The `chosen_or_highest_parent` field is set to the highest `chosen_or_highest_parent` of all received N-TIEs and S-TIEs for the tree, identifying the root of all sub-trees from that TOF node's point of view. The `sub_tof_children` list all of sub-TOF nodes that have chosen the root as parent.

If a sub-TOF node that is the root of a sub-tree receives from TOF neighbors some S-TIE for the same tree but with different `chosen_or_highest_parent` values, it chooses, from all its TOF neighbors that are recorded as a `chosen_or_highest_parent`, the one with the highest system-id and (re)parent to that neighbor if that neighbor is not already its parent.

After the above steps, if a TOF node remains as the chosen parent of

some sub-TOF nodes but its system-id does not match the highest chosen_or_highest_parent of all N-TIEs and S-TIEs (i.e. the root), the TOF node needs to join towards the root through some intermediate sub-TOF and TOF nodes. If it has a sub-TOF neighbor listed in the sub_tof_children of the root, it originates an S-TIE with the per-neighbor flooding scope set to the sub-TOF neighbor, i.e. the sub-TOF neighbor now becomes the parent of the TOF node (that is a parent of some other sub-TOF nodes).

In case the TOF node does not have a neighbor listed in the sub_tof_children of the S-TIE for the root, further study is needed. It could be that the topology is so partitioned that a spanning tree could not be built.

[3.](#) Security Considerations

To be provided.

[4.](#) Acknowledgements

The authors thank Bruno Rijsman and Antoni Przygenda for their review and suggestions.

[5.](#) References

[5.1.](#) Normative References

[I-D.ietf-rift-rift]

Przygienda, T., Sharma, A., Thubert, P., Rijsman, B., and D. Afanasiev, "RIFT: Routing in Fat Trees", [draft-ietf-rift-rift-12](#) (work in progress), May 2020.

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

[5.2.](#) Informative References

[I-D.zzhang-pim-pds]

Zhang, J. and K. Patel, "Protocol Dependent Multicast Signaling", [draft-zzhang-pim-pds-00](#) (work in progress), October 2015.

[RFC5015] Handley, M., Kouvelas, I., Speakman, T., and L. Vicisano, "Bidirectional Protocol Independent Multicast (BIDIR-PIM)", [RFC 5015](#), DOI 10.17487/RFC5015, October 2007, <<https://www.rfc-editor.org/info/rfc5015>>.

Authors' Addresses

Zhaohui Zhang
Juniper Networks

EMail: zzhang@juniper.net

Pascal Thubert
Cisco Systems, Inc

EMail: pthubert@cisco.com