rift
Internet-Draft
Intended status: Standards Track
Expires: 29 August 2026

Y. Zhou, Ed.
L. Zhu, Ed.
C. Wen, Ed.
China Unicom

25 February 2026

Bidirectional Forwarding Detection (BFD) for Routing in Fat Trees (RIFT) draft-zhou-rift-bfd-00

Abstract

This document specifies the use of Bidirectional Forwarding Detection (BFD) for fast failure detection of Routing in Fat Trees (RIFT) adjacencies. RIFT is specified in RFC 9692. While RFC 9692 describes interactions with BFD, it does not define normative behavior. This document specifies the use of single-hop BFD, as defined in RFC 5881, for RIFT adjacencies, including behavior for parallel links and multiple RIFT instances.

Table of Contents

1. Introduction

With large-scale AI and model-training workloads becoming commonplace in data center networks, transient network failures (e.g., link flaps, port flapping, and optical module/cabling anomalies) can create short-lived blackholes and introduce reroute latency. In practice, such impairments are amplified into application-level tail latency, job retries, and even reduced aggregate training throughput. As a result, the control plane of a fat-tree fabric is expected to provide sub-second and often millisecond-scale failure detection and convergence. RIFT's keepalive mechanism, centered around a seconds-scale holdtime, is inherently oriented toward stability and scalable adjacency maintenance, and is therefore not well suited to serve as the primary trigger for sub-second fast convergence. Consequently, it is necessary to introduce and standardize the use of Bidirectional Forwarding Detection (BFD) in RIFT-based fat-tree deployments in order to meet the fast failure-convergence requirements of training and similar latency-sensitive workloads.

This document specifies the use of single-hop BFD, as defined in [RFC5881], for failure detection of RIFT adjacencies. It defines BFD session establishment, interaction with the RIFT adjacency state machine, and behavior for parallel links and multiple RIFT instances.

1.1. Terminology

This document uses the acronyms defined in [RFC9692] along with the following:

RIFT instance: A logical instance of the RIFT protocol running on a node.

BFD session: A BFD control session as defined in [RFC5881].

1.2. Conventions Used in This Document

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.

2. Overview of BFD Usage in RIFT

Implementations of RIFT using BFD as specified in this document MUST use single-hop BFD as defined in [RFC5881]. BFD is used for rapid detection of failures that affect RIFT adjacencies. BFD operation is independent of RIFT ThreeWay hello and holddown timer.

3. BFD Session Establishment

RIFT advertises BFD capability on a per-link basis. In [RFC9692], support for BFD is indicated via the LinkCapabilities field carried in LIE messages. Based on this capability advertisement, an implementation MAY enable BFD to accelerate RIFT adjacency state updates and achieve faster routing convergence.

A BFD session SHOULD be started after the corresponding RIFT adjacency has reached ThreeWay state. If link identifiers or BFD capabilities change, both the LIE and any BFD sessions SHOULD be brought down and back up again. In case only the advertised capabilities change, the node MAY choose to persist the BFD session.

3.1. Parallel Link Considerations

Sharing a single BFD session across multiple parallel links may lead to incorrect liveness inference for individual links (e.g., partial link failures becoming invisible to the control plane, or being amplified into a full adjacency failure). Therefore, for parallel links between the same pair of RIFT nodes, each link is RECOMMENDED to have an independent BFD session.

With per-link BFD sessions, each parallel link can fail independently. When one or more (but not all) parallel links experience a BFD Down condition, the advertising node SHOULD update its Node TIE to reflect the affected link(s), for example by adjusting the corresponding link_ids and bandwidth attributes, without unnecessarily treating the entire adjacency as unreachable.

In deployments where precise per-link failure information is not required, implementations MAY share a single BFD session across multiple parallel links. However, this document does not specify procedures or interoperability requirements for shared-session operation.

3.2. BFD Parameters

After RIFT ThreeWay hello adjacency convergence, a BFD session MAY be formed automatically between the RIFT endpoints without further configuration using the exchanged discriminators that are equal to the local_id in the LIEPacket.

BFD sessions for RIFT MUST follow the encapsulation and demultiplexing rules defined in RFC 5881. To ensure that BFD provides meaningful acceleration of failure detection relative to the RIFT LIE keepalive mechanism, the negotiated BFD parameters (e.g., Desired Min TX Interval, Required Min RX Interval, and Detect Mult) SHOULD result in a BFD Detection Time that is significantly smaller than the LIE holdtime advertised on the adjacency (the default LIE FSM holdtime is 3 seconds in [RFC9719]).

4. Interaction with the RIFT

A BFD session transition to Down on a RIFT-enabled link can result in the following actions within the local RIFT instance:

1. The corresponding RIFT adjacency SHOULD be re-initialized (i.e., the LIE FSM is reset and restarted from its initial state), and any adjacency-related state and timers (e.g., holddown timer) are cleared as appropriate.

2. The node SHOULD update the neighbor information advertised in its Node TIE to reflect the loss of the affected adjacency/link(s).

3. The updated TIE information SHOULD be flooded according to the RIFT flooding procedures (e.g., via TIE/TIDE/TIRE exchanges).

4. Nodes receiving the updated topology information SHOULD perform reachability recomputation (e.g., N-SPF and/or S-SPF) as required.

5. If the resulting topology change satisfies the conditions for disaggregation, disaggregation procedures MAY be triggered (in particular for multi-plane "fallen leaf" scenarios).

Besides, in case an established BFD session goes Down after it was Up, RIFT adjacency SHOULD be re-initialized and subsequently started from Init after it receives a consecutive BFD Up.

5. Multi-Instance Considerations

In RIFT, a node may be configured with multiple RIFT instances. Such instances can be deployed over distinct interfaces, or, subject to local configuration, over the same physical, sub-, or logical interface. Each RIFT instance maintains its own adjacency state and control-plane state independent of other instances.

5.1. Shared BFD Model

A BFD session is associated with a specific point-to-point link, and its state transitions can be used as an input signal for any RIFT instance running over that link. Therefore, to reduce the total number of BFD sessions, implementations MAY share a single BFD session across multiple RIFT instances on the same interface (i.e., between the same pair of nodes over the same link). If a shared BFD session transitions to Down, all adjacencies of the associated RIFT instances that depend on that link MUST treat the link as failed (e.g., the adjacencies MUST be re-initialized or brought down, consistent with the implementation’s adjacency handling procedures).

When the shared-BFD model is used, all associated RIFT instances on the same interface MUST use compatible BFD parameters such that a single BFD session can be established and maintained. If incompatible requirements are configured (e.g., conflicting BFD interval and detection parameters that cannot be satisfied by one session), the implementation MUST either:

a. fall back to the per-instance BFD model, or

b. reject the configuration.

5.2. Per-Instance Model

Different RIFT instances on the same node may have different operational requirements for BFD (e.g., different failure-detection targets or stability thresholds). In such cases, the shared-BFD model cannot be used. A separate BFD session MUST be established for each RIFT instance adjacency.

In the per-instance BFD model, each BFD session is uniquely associated (by local demultiplexing policy) with:

* the underlying point-to-point link; and

* a specific RIFT instance adjacency on that link.

To establish multiple BFD sessions over the same interface, an implementation MAY use one or more of the following demultiplexing mechanisms:

a. distinct UDP source ports;

b. distinct discriminators. In this case, the discriminator allocation MUST NOT rely solely on the per-link identifier described in Section 3.2; instead, discriminators MUST be allocated such that they uniquely identify the combination of a RIFT instance and a link.

If a per-instance BFD session transitions to the Down state, only the corresponding RIFT instance adjacency MUST be treated as failed (e.g., brought down or re-initialized), and other RIFT instances on the same interface MUST NOT be affected.

5.3. Restart Behavior

When a RIFT instance is restarted or reconfigured, only the BFD sessions associated with that instance MUST be affected in the per-instance model. In the shared model, a restart of a single RIFT instance MUST NOT reset the shared BFD session unless all associated instances are removed.

6. Security Considerations

This document does not introduce new protocol mechanisms beyond those defined in [RFC5880], [RFC5881], and [RFC9692]. The security considerations described in those documents apply to the mechanisms specified here.

In particular, the security properties of BFD, including optional authentication, as described in [RFC5880], remain applicable. The single-hop protection mechanisms described in [RFC5881], such as the use of a TTL or Hop Limit of 255, also apply.

In the shared BFD session model, a failure, misconfiguration, or attack that causes the shared BFD session to transition to the Down state may result in the simultaneous loss of multiple RIFT adjacencies across different RIFT instances. Such correlated adjacency failures may lead to transient routing instability or increased convergence events within the affected topology. Operators SHOULD carefully evaluate the use of the shared BFD session model, taking into account scaling requirements and the potential impact of larger failure domains. Where strict failure isolation between RIFT instances is required, the per-instance BFD model defined in this document SHOULD be used.

7. IANA Considerations

TBD.

8. Normative References

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

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

[RFC5881] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD) for IPv4 and IPv6 (Single Hop)", RFC 5881, DOI 10.17487/RFC5881, June 2010, <https://www.rfc-editor.org/info/rfc5881>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[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, April 2025, <https://www.rfc-editor.org/info/rfc9692>.

[RFC9719] Zhang, Z., Wei, Y., Ma, S., Liu, X., and B. Rijsman, "YANG Data Model for Routing in Fat Trees (RIFT)", RFC 9719, DOI 10.17487/RFC9719, April 2025, <https://www.rfc-editor.org/info/rfc9719>.

Authors' Addresses

Yu Zhou (editor)
China Unicom
Beijing
China
Email: zhouy739@chinaunicom.cn

Lin Zhu (editor)
China Unicom
Beijing
China
Email: zhul14@chinaunicom.cn

Chenyang Wen (editor)
China Unicom
Beijing
China
Email: wency15@chinaunicom.cn