Routing Bridges (RBridges): Operations, Administration, and Maintenance (OAM) Support
draft-ietf-trill-rbridge-oam-02

Abstract

Routing Bridges (RBridges) implement the TRansparent Interconnection of Lots of Links (TRILL) protocol which provide a transparent least-cost frame routing in multi-hop networks with arbitrary topologies, while also inherently providing loop mitigation. As RBridges are deployed in real-world situations, operators will need tools for debugging problems that arise. This document specifies a set of RBridge features for operations, administration, and maintenance (OAM) purposes in RBridge campuses. The features specified in this document include tools for traceroute, ping, and error reporting.

Status of This Memo

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

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

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

This Internet-Draft will expire on September 13, 2012.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of
Table of Contents

1. Introduction ........................................ 4
   1.1. Requirements Language ............................ 4
2. Acronyms ............................................ 4
3. TRILL OAM Message .................................. 6
4. RBridge Tools ...................................... 7
   4.1. Application RBridge Tools ...................... 7
      4.1.1. RBridge Ping ................................ 8
      4.1.2. Hop Count Traceroute ....................... 9
         4.1.2.1. Path Sharing ............................ 10
         4.1.2.2. Multi-Destination Targets ................ 11
   4.2. Error Reporting ................................ 12
      4.2.1. Hop Count Zero Error ....................... 12
      4.2.2. MTU Error .................................. 13
5. RBridge Channel Message Format ..................... 13
   5.1. RBridge Channel Header and Sequence Number .... 13
6. OAM Protocol Field Values .......................... 14
   6.1. Response Frame Field Values .................... 14
   6.2. Self-Initiated Frame Field Values ............... 16
7. OAM Protocol Formats ................................ 17
   7.1. Protocol Application Codes Formats ............. 17
      7.1.1. Echo Request ............................... 17
      7.1.2. Echo Reply ................................ 18
   7.2. Error Notification Format ........................ 19
      7.2.1. Error Specifiers ............................ 20
8. Type, Length, Value (TLV) Encodings ................ 21
   8.1. Next Hop Information ............................ 23
   8.2. Previous Hop Information ....................... 24
   8.3. Incoming Port ID ................................ 24
   8.4. Outgoing Port ID ................................ 25
   8.5. Outgoing Port MTU ............................... 25
   8.6. IS-IS System ID ................................ 26
9. Acknowledgments ..................................... 26
10. IANA Considerations ................................ 26
11. Security Considerations ........................... 27
12. References ......................................... 27
   12.1. Normative References ........................... 27
   12.2. Informative References ........................ 28
Appendix A. Implementation Considerations ............... 29
   A.1. Hop Count Traceroute Example ................... 29
   A.2. Ping Example ................................... 31
Appendix B. Revision History ............................ 32
   B.1. Changes from -01 to -02 ....................... 32
   B.2. Changes from -00 to -01 ....................... 33
1. Introduction

The IETF has standardized RBridges, devices that implement the TRILL protocol, a solution for transparent least-cost frame routing in multi-hop networks with arbitrary topologies, using a link-state routing protocol technology and encapsulation with a hop-count [RFC6325]. As RBridges are deployed, operators will require tools for troubleshooting of operations issues in the network. TRILL uses IS-IS for the control plane [IS-IS] [RFC6165] [RFC6326]. IS-IS has a link-state database which contains the information of all links in the TRILL domain and IS-IS has a routing table. This information can be used for trouble shooting purposes.

There are a number of mechanisms to verify the control plane/data plane information, however correctness of the control plane information does not guarantee the data plane is working correctly. This motivates the need for OAM tools that allow an operator to test the data plane. Protocols such as IP, MPLS, and IEEE 802.1 have features enabling an operator to exercise the data plane [RFC4443] [RFC0792] [IEEE.802-1ag]. There is a need for a similar set of tools in TRILL. Likewise, there is a need for error reporting capabilities inside an RBridge campus.

Sometimes there may be a need for faster convergence than is provided by the TRILL hello protocol. Such fault notification functionality is not specified in this document. [BFD] provides this functionality using BFD.

This document specifies a set of RBridge features for operations, administration, and maintenance purposes in RBridge campuses along with the procedures and frame formats for these features. The features specified in this document include tools for traceroute, ping, and error reporting. Section 3 of this document specifies the general usage of a defined message format. Section 4 specifies some additional applications of the message format. Section 5 specifies the format and value of the messages on the wire.

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

2. Acronyms

- BPDU - Bridge PDU
- CHbH - Critical Hop-by-Hop
- CItE - Critical Ingress-to-Egress
- DA - Destination Address
- DR - Designated Router
- DRB - Designated RBridge
- ES - End Station
- ESa - End Station A
- ESb - End Station B
- ECMP - Equal-Cost Multi-Path
- ESADI - End Station Address Distribution Instance
- FCS - Frame Check Sequence
- ID - Identification
- IEEE - Institute of Electrical and Electronics Engineers
- IETF - Internet Engineering Task Force
- IP - Internet Protocol
- IS-IS - Intermediate System to Intermediate System
- MAC - Media Access Control
- MPLS - Multiprotocol Label Switching
- MTU - Maximum Transmission Unit
- OAM - Operations, Administration, and Maintenance
- P2P - Point-to-point
- PDU - Protocol Data Unit
- RBridge - Routing Bridge
- SA - Source Address
To facilitate message passing as needed by the OAM requirements, the TRILL RBridge Channel facility \([\text{RBridgeChannel}]\) is utilized.

There are two types of TRILL OAM messages defined in this document carried within an RBridge Channel: application and error notification. Frames with an error notification MUST NOT be generated in response to frames that are an error notification. Implementations SHOULD rate limit the origination of error notifications. Whereas unknown unicast frames are sent as multi-destination messages, sending unknown unicast frames with an error can lead to an amplification attack. As such special care and rate limiting are necessary for error notifications.

Error notification messages contain the error-causing frame or the initial part thereof after its OAM message. The following are two figures showing application and error notification message structure. Section 5 goes into the details of these formats.

---

**Application Frame**

```
+----------------------------+
|     Outer Link Header      |
+----------------------------+
|        TRILL Header        |
+----------------------------+
|   Inner Ethernet Header    |
+----------------------------+
|   RBridge Channel Header   |
+----------------------------+
| OAM Protocol Spec. Payload |
+----------------------------+
```

---
RBridge campuses do not, in general, guarantee lossless transport of frames so a frame containing a TRILL OAM Message, possibly generated in response to some other frame, might be lost.

4. RBridge Tools

This section specifies a number of RBridge OAM tools. For classification purposes they are divided into two sections, applications and error tools. Both of these tools use messages called echo requests and echo replies. The format is described in Section 5. An echo request is a message that says please respond. The echo reply is that response. The exact usage is further defined in this section. These messages also contain TLV fields which carry additional information in regards to the message. The formats of these TLVs are described in Section 8.

4.1. Application RBridge Tools
4.1.1. RBridge Ping

Ping is a tool for verifying RBridge connectivity. The ping-originating RBridge transmits one or more TRILL data frames with a TRILL OAM message. This message contains the code of an echo request (See Section 7.1.1). The ingress RBridge MUST be the frame-originating RBridge. The egress RBridge is the destination RBridge to which connectivity will be checked. The M bit MUST be zero.

The purpose of the ping is to confirm connectivity of the data plane, and options defined in future drafts MAY be included. The purpose of allowing the addition of options is so that the frame mimics a data frame that follows the same path through the data plane that a ‘real’ data frame would. An RBridge Ping, however, uses the OAM Channel and so depending on the ECMP hashing used by RBridges in the campus it may not in fact share the same path as ‘real’ data going through the network. The traceroute tool has a way to ensure the data follows the same path as the data does and if an operator wishes to test that path the data takes, the traceroute functionality ought to be used.

The echo request MAY have an arbitrary 28-bit unsigned integer sequence number to assist in matching reply messages to the request. In most circumstances, a single echo request is needed to complete the ping but it might be desirable for a single RBridge to ping multiple egress RBridges, or trace differing flows simultaneously. Assigning differing sequence numbers to each frame aids in matching which trace the reply belongs to.


RBridges implementing ping SHOULD issue a reply in response to this request. See Section 11 for reasons that RBridges are allowed to choose not to respond to a request. If an RBridge chooses to respond to the request, the reply MUST consist of one TRILL data frame per request with an OAM message containing the protocol code of an echo reply. The echo reply MUST have the same sequence number as the request being matched.

For the echo reply the ingress RBridge field MUST be the reply-originating RBridge’s nickname. The egress RBridge MUST be the request-originating RBridge’s nickname. The Inner.VLAN, Inner.MacSA, and Inner.MacDA SHOULD default to the values specified in Section 6.1. The M bit MUST be zero for a known unicast ping.

The reply-originating RBridge SHOULD include its 16-bit port ID from the port on which the request was received in the incoming port field of the reply. It SHOULD also include its 16-bit port ID from the

Bond & Manral          Expires September 13, 2012               [Page 8]
port on which the frame would be forwarded. A port ID of 0xFFFF indicates the frame would not have been forwarded and that the frame was consumed by the RBridge itself.

The reply frame need not follow the same path though the campus as the request. The reply messages are not meant to test the data plane.

End stations are not involved in this the ping process. RBridge pings are from RBridge to RBridge. While the frames sent may emulate data sent from ESa to ESb, the end stations are not, in fact, involved.

The transmitting RBridge MUST wait for a reply frame until a time-out occurs. At that time, the RBridge SHOULD assume the frame was lost, and this SHOULD be indicated to the operator. The length of this time-out is beyond the scope of this document.

4.1.2. Hop Count Traceroute

The ability to trace the path the data takes through the network is an invaluable debugging tool. RBridge traceroute provides this functionality through use of the TRILL OAM message (See Section 3). In a hop-count traceroute, the originating RBridge starts by transmitting one TRILL data frame with a TRILL OAM message. This message contains a protocol code of an echo request (See Section 7.1.1). The ingress RBridge MUST be the RBridge originating the frame.

When a traceroute is initiated, it is either targeting a known unicast target or a multi-destination target as specified by the operator. If the hop-count traceroute is for a known unicast target, the egress RBridge is the destination RBridge to which connectivity will be checked and the M bit MUST be zero. Otherwise, if the hop-count traceroute is for a multi-destination target, the egress RBridge is the distribution tree nickname for the traceroute. Multi-destination targets are handled the same as known unicast targets but require a small amount of additional logic as specified in Section 4.1.2.2.

The first echo request frame transmitted MUST have a hop-count of zero. The RBridge will continue transmitting these echo requests, incrementing the hop-count by one each time until a hop-count error notification from the destination nickname as its ingress nickname is received. If a transit RBridge decrements the hop-count by more than one, it MAY transmit multiple hop-count error notifications.

The purpose of the traceroute is to confirm connectivity of the data
plane, and therefore options defined in future drafts MAY be included. The purpose of allowing the addition of options is so that the frame mimics a data frame that follows the same path through the data plane that a 'real' data frame would. The ability to share the same path as 'real' data is further specified in Section 4.1.2.1.

The echo request MAY have an arbitrary 28-bit unsigned integer sequence number to assist in matching reply messages to the request. This is important for the hop-count traceroute since replies may return to the ingress RBridge in a different order then their matching requests were sent.


The replying RBridge SHOULD include its 16-bit port ID from the port on which the hop-count error generating frame was received in the Incoming Port ID TLV of the reply. It SHOULD also include its 16-bit port ID from the port on which the frame would be forwarded if the frame did not have a hop-count error in the Outgoing Port ID TLV. A port ID of 0xFFFF indicates the frame would not have been forwarded and would be consumed by the RBridge itself. Finally the reply SHOULD include a 16-bit nickname and 48-bit system id of the next hop RBridge the frame would have been sent to if there were no error in the Next Hop Nickname TLV. If this RBridge is the egress RBridge this TLV MUST NOT be included in the response. This is to facilitate knowledge of a more precise path through the campus as seen in RFC 5837 [RFC5837].

The advantage of this traceroute method is that the transit RBridges do not have to do any special processing of the frames until a hop-count error is detected, a condition they are required to detect by the TRILL base protocol. The disadvantage is the request-originating RBridge needs to transmit as many frames as there are hops between itself and the destination RBridge.

The end stations are not involved in this process. RBridge traceroutes are from RBridge to RBridge. While the frames sent may emulate data sent from ESa to ESb, the end stations are not, in fact, involved. An Rbridge must keep the TRILL header contents the same for ever frame sent in a hop count traceroute.

4.1.2.1. Path Sharing

In certain cases it could be important to send 'real' data over a network as to test the path that 'real' data takes and to test the fate that such real data would have. Simple sending an RBridge channel message is insufficient because many RBridge implementations
will use various forms of ECMP hashing based on fields such as MAC addresses, IP addresses, and/or TCP/UDP port numbers. To satisfy this need for path sharing an RBridge originating a traceroute MAY send a data packet instead of an echo request. The data packet will look entirely like an encapsulated data frame, with whatever fields the user specifies to ensure path sharing. The one exception is that the hop count will be set as described previously: incremented as the traceroute proceeds. Since these frames will not include a sequence number, these data frames must be sent in lock step: waiting for a timeout or an hop count error before sending the next incremented hop count frame. Since this data frame looks like a real frame but is in fact not real, when the egress RBridge is reached in the traceroute the originating RBridge MUST NOT send trace frames with higher hop-counts. RBridge ping does not have an equivalent path sharing mechanism since it tests end to end connectivity rather than the exact path taken.

4.1.2.2. Multi-Destination Targets

For multi-destination targets at each branch in the tree the tagged frame will be replicated causing each RBridge in the tree, possibly pruned by VLAN and/or IP multicast group, to send a response to the echo request. If all RBridges in the possibly pruned distribution tree support the echo request message, then the ingressing RBridge will receive an error notification from each of them. These error replies are staggered by distance from the generating RBridge. Meaning the first set of responses come from the first request send with hop count equal to zero and these replies will be from this RBridges neighbors. The second set of responses will come from RBridge two hops away and so forth.

The ingressing RBridge can compile all of these notifications, using the parent pointers located in the previous hop information TLV, into an output of the tree the traffic traversed. A traceroute application SHOULD report any errors received, such as an invalid distribution tree nickname, caused by the hop-count traceroute frames. RBridges receiving a multicast destination echo request MUST NOT transmit an echo reply if the multi-destination bit is set. Echo requests that are not used with the hop-count traceroute come from the ping tool, and ping messages are not valid as multi-destination traffic. In a hop count traceroute, devices will already be transmitting a hop-count error notification and so there is no reason to transmit a double set of replies. A multi-destination hop-count traceroute stops when the transmitted hop count reaches the maximum, 0x3F. One cannot use the diameter of the network to limit when this traceroute stops because some RBridges may decrement the hop count by more than one.
In multi-destination request frames, the Previous Information TLV MUST be set to the nickname and system id of the RBridge the frame was received from. This is the previous hop RBridge. The Next Hop Information TLV is not used in multi-destination traceroute frames.

4.2. Error Reporting

Errors can occur in received TRILL data frames. For this purpose, the error notification format is specified. These are generated due to various events as specified subsequently. When a TRILL data frame is received with an error, an error notification frame SHOULD be generated. See Section 11 for reasons some RBridges are allowed to choose not to respond to a request. The generated reply MUST contain the error notification. The sub-code MUST contain a code specifying the error encountered. The valid sub-code values are specified in Section 7.2.1. Two of these sub-codes provide for TLVs with additional information. The error notification also contains a 3 bit error type field which describes the error.

This frame has a TRILL header and it contains, as its payload, the frame received with the error. If the size of the received frame would cause the generated frame to exceed 1470 bytes, the frame MUST be truncated to the 1470 bytes. The payload MUST include the TRILL header of the received frame and MUST NOT include the link-layer header. The generated reply MUST contain the error notification message specific to the error.

When the original ingress RBridge receives the error frame, at a minimum, the RBridge SHOULD update a counter specifying the number of error frames received for the causing error. The encapsulated frame MUST NOT be egressed.

The two sub-codes that provide for TLVs with additional information are described below. All other sub-codes specified in this document do not normally contain TLVs.

4.2.1. Hop Count Zero Error

When a TRILL data frame is received with a hop-count of zero, an error notification frame SHOULD be generated unless rate limiting or some particular difficulty, as described below, stops the sending of such an error notification. The generated reply MUST contain the hop-count zero error sub-code. If the received frame has the echo request message, the hop-count zero error notification MUST have a sequence number matching the echo request. Otherwise, the sequence number MUST be set to zero. The Incoming Port ID TLV SHOULD be included with the port ID the received frame arrived on. The Outgoing Port ID TLV SHOULD be included with the port ID of the port.
the received frame would have been forwarded onto if the hop-count was not zero. Finally, the error notification SHOULD include a 16-bit nickname and 48-bit system id of the next hop RBridge the frame would have been sent to in the Next Hop Information TLV. If the request is a multi-destination frame, the previous hop information SHOULD be included instead with it set to the nickname and system id of the RBridge the frame was received from. This is the previous hop RBridge. If the RBridge transmitting the reply is the egress RBridge, this TLV MUST NOT be included in the frame.

4.2.2. MTU Error

When a TRILL data frame is received with a payload that would exceed the MTU of the port the frame would otherwise be forwarded to, an error notification frame MAY be generated. The generated reply MUST contain the MTU error sub-code. The Outgoing Port MTU TLV MUST be included with the MTU of the port the frame would have otherwise been transmitted on. The Incoming Port ID TLV SHOULD be included with the port ID the received frame arrived on. The Outgoing Port ID TLV SHOULD be included with the port ID of the port the received frame would have been forwarded onto if the frame size was not too large. Finally, the error notification message SHOULD include a 16-bit nickname and 48-bit system id of the next hop RBridge the frame would have been sent to in the Next Hop Information TLV. If the request is a multi-destination frame, the previous hop information SHOULD be included instead with it set to the nickname and system id of the RBridge the frame was received from. This is the previous hop RBridge. If the RBridge transmitting the reply is the egress RBridge, this TLV MUST NOT be included in the frame.

5. RBridge Channel Message Format

This section specifies the format of the TRILL OAM payload after the RBridge Channel header and values of the fields in the RBridge Channel Header [RBridgeChannel].

5.1. RBridge Channel Header and Sequence Number

The RBridge Channel Header [RBridgeChannel] fields and flags and following sequence number are as follows:

- CHV (Channel Header Version) is zero.
- Protocol code values are:
  - 0x004 (Suggested): Echo
0x005 (Suggested): Error Notification

- Flags: The SL and NA bits SHOULD be zero, the MH bit SHOULD be one
- ERR: The ERR field MUST be zero.

- SPID and Sequence Number: For the Echo and Error Notification protocols, the RBridge Channel Header is always followed by a nibble sub-protocol identifier (SPID) and a 28-bit Sequence Number. This 28-bit field is used to sequence or match frames for certain uses. The SPID is used to provide additional op-code room for a protocol to further multiplex its messages. Not all TRILL OAM messages utilize the sequence number field or the SPID.

6. OAM Protocol Field Values

6.1. Response Frame Field Values

Frames with a TRILL OAM message generated in response to another TRILL data frame have fields set as follows unless otherwise specified:

- Frames of type Application or Error
  - Field: Inner.MacSA
  - Value: If the Inner.MacDA of the received frame is one of the MAC addresses of the RBridge generating the frame, the value MUST be that MAC address. Otherwise, it MUST be one of the RBridge’s MAC addresses.

- Frames of type Application or Error
  - Field: Inner.MacDA
  - Value: The value MUST be All-Egress-RBridges.

- Frames of type Application or Error
  - Field: Inner.VLAN ID
  - Value: If the frame is generated in response to another frame with a legal Inner.VLAN ID, it MUST be the Inner.VLAN ID of the received frame. In other cases, it SHOULD be the default VLAN ID 1.
Frames of type Application or Error

* Field: Ingress RBridge nickname

  Value: If the egress RBridge nickname of the received frame is a nickname of the RBridge generating the frame, then the value MUST be that nickname. otherwise, it MUST be one of the RBridge’s nicknames.

Frames of type Application or Error

* Field: Egress RBridge nickname

  Value: The value MUST be the ingress RBridge nickname of the received frame. Except that, if the ingress RBridge nickname received is unknown or reserved the frame MUST be generated on the port the frame was received on with an Outer.MacDA and egress RBridge nickname of the previous-hop RBridge if this is known.

Frames of type Error

* Field: Offending Encapsulated Frame

  Value: The value MUST be N bytes of the frame that had the error where N is the minimum of the frame size and the number of bytes that would bring the resulting error frame up to 1470 bytes. This MUST include the TRILL header and MUST NOT include the link-layer header.

Frames of type Application

* Field: M Bit

  Value: The value of this field is defined by each specific OAM protocol.

Frames of type Error

* Field: M Bit

  Value: The value MUST be zero.

Frames of type Application or Error

* Field: Inner.Priority
* Value: The value SHOULD be one less than the priority of the received frame, but not less than the lowest priority. One less may be numerically one less in the normal case or logically one less in the case of priority mapping being present.

6.2. Self-Initiated Frame Field Values

Frames with a TRILL OAM message that are self-initiated have fields set as follows unless otherwise specified:

- Frames of type Application
  - Field: Inner.MacSA
    - Value: This SHOULD be one of the transmitting RBridge’s MAC addresses. The Inner.MacSA MAY be other values as specified in Appendix A.

- Frames of type Application
  - Field: Inner.MacDA
    - Value: The value SHOULD be All-Egress-RBridges.

- Frames of type Application
  - Field: Inner.VLAN ID
    - Value: The value SHOULD be the default VLAN ID 1. The Inner.VLAN ID MAY be other values as specified in Appendix A.

- Frames of type Application
  - Field: Ingress RBridge nickname
    - Value: The value SHOULD be one of the RBridge’s nicknames. The Ingress RBridge nickname MAY be other values as specified in Appendix A.

- Frames of type Application
  - Field: Egress RBridge nickname
    - Value: The value of this field is defined by each specific OAM protocol.
Frames of type Application

* Field: M Bit

* Value: The value of this field is defined by each specific OAM protocol.

Frames of type Application

* Field: Inner.Priority

* Value: The value of this field defaults to zero. The Inner.Priority MAY be other values as specified in Appendix A.

7. OAM Protocol Formats

The formats of Echo Request, Echo Reply, and Error Notification OAM Messages are given below.

7.1. Protocol Application Codes Formats

7.1.1. Echo Request

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RBridge Channel</td>
<td></td>
<td>Header</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SPID</td>
<td></td>
<td>Sequence</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Echo Request

Figure 3

This message is used by ingress R Bridges to request an echo reply from the egress R Bridge. Further uses are specified in Section 4.1.2 and Section 4.1.1

- SPID: The SPID MUST be zero to indicate an echo request.

- Sequence Number: An arbitrary 28-bit unsigned integer used to aid in matching reply messages to echo requests. It MAY be zero.
7.1.2. Echo Reply

This message is used by egress RBridges to reply to an echo request from the ingress RBridge. Further uses are specified in Section 4.1.2 and Section 4.1.1.

- SPID: The SPID MUST be one to indicate an echo reply.
- Sequence Number: A 28-bit unsigned integer used to aid in matching reply messages to echo requests. Set to the sequence number field of the Echo Request that cause this Echo Reply.
- TLVs: A set of type, length, value encoded fields as specified in Section 8.
7.2. Error Notification Format

```
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |10|11|12|13|14|15|
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|                RBridge Channel                |
|                Header                          |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
| SPID  |    Sequence        |
|       |    Number          |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
| Err. T. | Subcode          |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
   .            .             .
   .            .             .
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+

Error Format

Figure 5
```

This message is used by RBridges to signal that an error has been detected.

- **SPID**: The SPID MUST be set to all zeros on transmission and is ignored on reception. It is unused by the error notification protocol.

- **Sequence Number**: For all sub-codes except for the hop count error this field is unused. It is set to zero on transmission and ignored on reception. For the hop count error this is a 28-bit unsigned integer used to aid in matching reply messages to echo requests. If the frame whose hop-count dropped to zero contains the echo request message (See Section 7.1.1), this MUST match the sequence number Echo Request found in that message. If this is not in reply to an Echo Request, then the sequence number MUST be set to zero.

- **Error Type**: MUST be a specifier of the error type describing the error. The values are: 0 (Permanent Error), 1 (Transient Error), 2 (Warning), 3 (Comment). Values 4 through 7 are available for allocation by IETF Review.

- **Subcode**: MUST be a specifier of the error discovered in the frame. The valid values are specified in Section 7.2.1
7.2.1. Error Specifiers

The sub-code values fall into three categories: errors (divided into transient and permanent errors), warnings, and comments. All sub-codes represent something out of the ordinary that has gone wrong, but certain ones are more important than others. Sub-codes that are classified as errors are the most severe with warning sub-codes being less severe. These are enabled by default. Errors can be further divided into transient and permanent. Transient errors are errors that happen but where the error causing RBridge can try again in the future and the error may not happen again. Permanent errors are errors that will happen again in a converged network. It is up to implementations to determine if errors should be listed as permanent or transient. Sub-codes classified as comments are minor and are disabled by default. They may be useful for operators debugging a network. All error generations are optional and therefore MAY be generated or not generated depending on security and implementation constraints.

The error specifiers sub-code values are:

Error Sub-codes

- 0: Unknown Error: Indicates an error has occurred.
- 1: Invalid Outer.VLAN: Indicates the Outer.VLAN ID was not the designated VLAN ID or was 0xFFFF.
- 2: Unknown Egress RBridge: Indicates the Egress RBridge in a received frame is unknown.
- 3: Unknown Ingress RBridge: Indicates the Ingress RBridge in a received frame is unknown. (RBridges are not required to test for this error.)
- 4: Unsupported Critical Hop-by-hop Option: Indicates an unsupported critical hop-by-hop option was received.
- 5: Unsupported Critical Ingress-to-Egress Option: Indicates an unsupported critical ingress-to-egress option was received.
- 6: Hop Count Zero: Indicates a frame hop count reached zero in transit. (Used for pings and traceroute.)
8. **Type, Length, Value (TLV) Encodings**

To facilitate future interoperable expansion of the data carried in OAM sub-messages some sub-messages use a TLV encoding. These TLV sections consist of a list of type, length, value encoded data where the type signals to the RBridge how to interpret the value, and the length tells the RBridge the length of the value in bytes. The type and length are both 16 bit fields. A length of zero indicates the value is a UTF-8 string with a NULL (‘\0’) terminating byte. Preceding the list of TLVs is a 16 bit total length field which specifies the total length of all the length fields in octet units. TLVs with an unknown Type MUST be ignored and skipped over. The value field is 1 byte aligned.
Each TLV in the TLV List appears on the wire encoded as follows:

```
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |10|11|12|13|14|15|
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|                     Type                      |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|                    Length                     |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
.                                               .
.                    Value                      .
.                   .                            .
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
```

TLV Format

Figure 7

The type values are:

- 0: Next Hop Information, See Section 8.1
- 1: Previous Hop Information, See Section 8.2
- 2: Outgoing Port ID, See Section 8.4
- 3: Incoming Port ID, See Section 8.3
- 4: Outgoing Port MTU, See Section 8.5
- 5: IS-IS System ID, See Section 8.6
8.1. Next Hop Information

```
+----------------------------------+
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
+----------------------------------+
|                                 |
+----------------------------------+
| 0x00                            |
+----------------------------------+
|                                 |
+----------------------------------+
| 0x08                            |
+----------------------------------+
| Next Hop Nickname                |
+----------------------------------+
|                                 |
+----------------------------------+
|                                 |
+----------------------------------+
| Next Hop System ID               |
+----------------------------------+
```

Next Hop Information Format

Figure 8

For traceroutes targeting known unicast destinations, hop-count errors, and MTU errors, this TLV MUST be a 16-bit nickname and 48-bit system ID of the next hop RBridge the frame is being or would have been sent to. If the next hop RBridge has not reserved a nickname the nickname field must be 0x0000. If the RBridge transmitting the TLV is the egress RBridge this TLV is not included in the frame. For pings, this field MUST be set to all zeros on transmission and ignored on reception. If an RBridge has multiple nicknames it SHOULD use the numerically largest nickname in the Next Hop Information TLV.
8.2. Previous Hop Information

For traceroutes targeting known unicast destinations, hop-count errors, and MTU errors, this TLV MUST be a 16-bit nickname and 48-bit system ID of the previous hop RBridge the frame being responded to was forwarded from. If an RBridge has multiple nicknames it SHOULD use the numerically largest nickname in the Previous Hop Information TLV.

8.3. Incoming Port ID

This TLV MUST be set to the Port ID found in ‘The Special VLANs and Flags sub-TLV’ for the port the request being replied to was received on [RFC6326].
8.4. Outgoing Port ID

This TLV MUST be set to the Port ID found in ‘The Special VLANs and Flags sub-TLV’ for the port the frame is being forwarded on to (or would have been for an echo request/hop-count error) [RFC6326]. If the request was consumed by the replying RBridge, the port ID MUST be 0xFFFF.

8.5. Outgoing Port MTU

This TLV MUST be the MTU of the outgoing port specified in the outgoing port ID TLV.
8.6. IS-IS System ID

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |10|11|12|13|14|15|
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
<table>
<thead>
<tr>
<th>Type = 0x04</th>
</tr>
</thead>
<tbody>
<tr>
<td>+--------------------------------------------</td>
</tr>
<tr>
<td>Length = 0x06</td>
</tr>
<tr>
<td>+--------------------------------------------</td>
</tr>
<tr>
<td>IS-IS System ID</td>
</tr>
<tr>
<td>+--------------------------------------------</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>IS-IS System ID Format</td>
</tr>
</tbody>
</table>

Figure 13

This TLV MUST include the IS-IS System ID of the system generating the message. This TLV MAY be included in all/any messages.

9. Acknowledgments

Many people have contributed to this work, including the following, in alphabetic order: Sam Aldrin, Dinesh Dutt, Donald E. Eastlake 3rd, Anoop Ghanwani, Meenakshi Kaushik, Jeff Laird, Thomas Narten, Santosh Rajagopalan, Marc Sklar, and Li Yizhou.

10. IANA Considerations

IANA is request to create a new subregistry within the TRILL Parameter registry for "TRILL OAM Message Error Sub-Message Error Specifiers". This subregistry that is initially populated as specified in Section 7.2.1. Additional values are allocated by IETF Review [RFC5226].

IANA is requested to create a new subregistry within the TRILL Parameter registry for "TRILL Error Reporting Protocol TLV Types" with initial values as listed in Section 5.3. Additional values are allocated by IETF Review [RFC5226].

This draft also requires action to reserve the TRILL RBridge Channel protocol codes. IANA is requested to allocate the TRILL RBridge Channel protocol codes for as listed in Section 5.1.
11. Security Considerations

The nature of the OAM Messages can lead to security concerns. By providing information about the topology and status of a network, attackers can gain greater knowledge of a network in order to exploit the network. Passive attacks such as reading frames with an OAM message could be used to gain such knowledge or active attacks where an attacker mimics an RBridge can be used to probe the network. Authentication, data integrity, protection against replay attacks, and confidentiality for TRILL OAM frames may be provided using a to-be-specified TRILL Security Option. Using such a security option would mitigate both the passive and active attacks.

For instance, data origin authentication could be provided in the future using a security options in the TRILL Header by verifying a hash using shared keys or a mechanism like SEND with CGA. To prevent replay attacks rate limiting, sequence numbers as well as some nonce based mechanism could be provided. Confidentiality for TRILL OAM frames could be provided based on some future security option extension which encrypts TRILL frames.

In a network where one does not wish to configure a security option, the threat of attackers is still present. For this reason, generation of any TRILL OAM Message frames is optional and SHOULD be configurable by an operator on a per RBridge basis. An RBridge MAY have this configurable on a per port basis. For instance, an operator SHOULD be able to disable hop-count traceroute reply messages or error notification message generation per port.

Another security threat is denial of service through use of OAM messages. For this reason, RBridges MUST rate limit the generation of OAM message frames. For multi-destination frames, the frames MAY be discarded silently to prevent any denial of service attacks in case of an error packet such as an 'options not recognized' error notification.

12. References

12.1. Normative References


12.2. Informative References


Appendix A. Implementation Considerations

This appendix contains a few considerations implementors should take note of when creating their user interface as well as some examples of what occurs when a traceroute or ping are executed. These provide a sample user interface one can use as the basis for their user interface.

First, an RBridge SHOULD maintain counters for each type of error generated. There SHOULD be a way for users to view these counters.

Some of the set of default field values for self originated frames are presented in Section 6.2. RBridges SHOULD be configurable to change these values to assign the TRILL data frame to a flow.

A.1. Hop Count Traceroute Example

Figure 14 contains a campus with three RBridges. Consider a hop-count traceroute from RB0 to RB2.

```
+-----+  +-------+   +-------+   +-------+  +-----+
| ESa +--+  RB0  +---+  RB1  +---+  RB2  +--+ ESb |
+-----+  |ingress|   |transit|   |egress |  +-----+
```

<table>
<thead>
<tr>
<th>Time</th>
<th>RB0</th>
<th>RB1</th>
<th>RB2</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1)</td>
<td>&lt;------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(2)</td>
<td></td>
<td></td>
<td>&lt;------</td>
</tr>
<tr>
<td>(3)</td>
<td></td>
<td></td>
<td>&lt;------</td>
</tr>
<tr>
<td>(4)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Hop Count Traceroute Example Topology

Figure 14
In this diagram RB0 transmits frame (1) destined to RB2. This frame contains the echo request message and a hop-count of 0. When RB1 receives this frame it drops it and transmits a hop-count-exceeded message, (2), to RB0. RB0 then transmits a frame, (3), with a hop-count of 1. RB1 decrements this hop-count by 1 to 0 and forwards it to RB2. RB2 drops frame (3) and transmits a Hop Count Zero error notification, (4), to RB0. The traceroute is now complete.

Below are some select fields for the frames:

<table>
<thead>
<tr>
<th>Frame #</th>
<th>Ingress RBridge</th>
<th>Egress RBridge</th>
<th>TRILL OAM Protocol</th>
<th>Sequence Number</th>
<th>Hop Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1)</td>
<td>RB0</td>
<td>RB2</td>
<td>Echo Request</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>(2)</td>
<td>RB1</td>
<td>RB0</td>
<td>Hop Count Zero</td>
<td>1</td>
<td>Default</td>
</tr>
<tr>
<td></td>
<td>RB1</td>
<td>RB0</td>
<td>error</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RB0</td>
<td>RB2</td>
<td>notification</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(3)</td>
<td>RB0</td>
<td>RB2</td>
<td>Echo Request</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>(3) @</td>
<td>RB0</td>
<td>RB2</td>
<td>Echo Request</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>RB0</td>
<td>RB2</td>
<td>Hop Count Zero</td>
<td>2</td>
<td>Default</td>
</tr>
<tr>
<td></td>
<td>RB0</td>
<td>RB2</td>
<td>error</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(4)</td>
<td>RB2</td>
<td>RB0</td>
<td>Hop Count Zero</td>
<td>2</td>
<td>Default</td>
</tr>
<tr>
<td>(4) @</td>
<td>RB2</td>
<td>RB0</td>
<td>error</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RB2</td>
<td>RB0</td>
<td>notification</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1: Hop Count Traceroute Example Frames

For example, if the nicknames for RB0, RB1, and RB2 are 0x1111, 0x2222, and 0x3333 respectively, the console output from such a trace might be:
Hop Count Tracing

<table>
<thead>
<tr>
<th>RBridge Incoming Port Id</th>
<th>Outgoing Port Id</th>
<th>RBridge Nexthop</th>
<th>Nickname</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x1111</td>
<td>Ingress</td>
<td>0x0001</td>
<td>0x2222</td>
</tr>
<tr>
<td>0x2222</td>
<td>0x0000</td>
<td>0x0001</td>
<td>0x3333</td>
</tr>
<tr>
<td>0x3333</td>
<td>0x0000</td>
<td>Egress</td>
<td>0x0000</td>
</tr>
</tbody>
</table>

Table 2: Hop Count Traceroute Example Output

In this example, the first line of output is generated from local information, no hop-count frames are sent to generate it.

A.2. Ping Example

Figure 15 contains a campus with three RBridges. Consider a ping from RB0 to RB2.

```
+-----+  +-------+   +-------+   +-------+  +-----+
| ESa +--+  RB0  +---+  RB1  +---+  RB2  +--+ ESb |
+-----+  |ingress|   |transit|   |egress |  +-----+

Time       RB0         RB1         RB2
.         (1)-------> (1) -------> |
.         | <------- (2) <-------(2)

Ping Example Topology

Figure 15
```

In this diagram RB0 transmits frame (1) destined to RB2. This frame contains the echo request message. When RB1 receives this frame it forwards it to RB2. When RB2 receives this frame it transmits and echo reply frame (2) destined to RB0. RB1 receives this frame and forwards it to RB0.

Below are some select fields for the frames:
For example, if the nicknames for RB0, RB1, and RB2 are 0x1111, 0x2222, and 0x3333 respectively, the console output from such a ping might be:

Pinging
---------------------
... from 0x1111 to 0x3333... 0x3333 is alive
... from 0x1111 to 0x3333... 0x3333 is alive
... from 0x1111 to 0x3333... 0x3333 is alive

In this example, the ping was repeated three times with the sequence number (not shown) being changed each time.

Appendix B. Revision History

RFC Editor: Please delete this appendix before publication.

B.1. Changes from -01 to -02

Moved the values table to the message format section and converted from table to list.

Added previous hop information TLV.

Removed error codes that were not needed.

Added path sharing traceroute with ‘real’ data being sent.

Added mention of BFD draft.

Made most TLVs optional to allow hardware/fast path implementations where this information might not be available.
Changed Next Hop Nickname TLV into Next Hop Information TLV since next hop might not always reserve a nickname. The new TLV includes the next hop system id.

Numerous minor typo corrections and wording clarifications.

B.2. Changes from -00 to -01

Broke down the table "frame field values" into two tables, "response frame field values" and "self initiated frame field values".

Reorganized the document to move user interface related items to the appendix and switched the order of ping/traceroute.

Numerous minor typo corrections and wording clarifications.

Authors’ Addresses

David Michael Bond
International Business Machines
2051 Mission College Blvd.
Santa Clara, CA  95054
US

Phone: +1-603-339-7575
EMail: mokon@mokon.net
URI: http://mokon.net

Vishwas Manral
Hewlett-Packard Co.
19111 Pruneridge Ave.
Cupertino, CA  95014
US

Phone: +1-408-447-0000
EMail: vishwas.manral@hp.com