INTERNET-DRAFT
Intended status: Proposed Standard

Donald Eastlake
Linda Dunbar
Huawei
Radia Perlman
EMC
Igor Gashinsky
Yahoo
Yizhou Li
Huawei

Expires: December 19, 2015
June 20, 2015

TRILL: Edge Directory Assist Mechanisms
<draft-ietf-trill-directory-assist-mechanisms-03.txt>

Abstract

This document describes mechanisms for providing directory service to TRILL (Transparent Interconnection of Lots of Links) edge switches. The directory information provided can be used in reducing multi-destination traffic, particularly ARP/ND and unknown unicast flooding. It can also be used to detect traffic with forged source addresses.

Status of This Memo

Table of Contents

1. Introduction..3
1.1 Uses of Directory Information..........................3
1.2 Terminology..4

2. Push Model Directory Assistance Mechanisms.............6
2.1 Requesting Push Service................................6
2.2 Push Directory Servers.................................6
2.3 Push Directory Server State Machine...................7
2.3.1 Push Directory States................................8
2.3.2 Push Directory Events and Conditions................9
2.3.3 State Transition Diagram and Table.................10
2.4 Additional Push Details...............................12
2.5 Primary to Secondary Server Push Service.............13

3. Pull Model Directory Assistance Mechanisms............14
3.1 Pull Directory Message Common Format.................15
3.2 Pull Directory Query and Response Messages...........16
3.2.1 Pull Directory Query Message Format................16
3.2.2 Pull Directory Responses...........................19
3.2.2.1 Pull Directory Response Message Format...........19
3.2.2.2 Pull Directory Forwarding........................21
3.3 Cache Consistency.....................................22
3.3.1 Update Message Format..............................25
3.3.2 Acknowledge Message Format.........................26
3.4 Summary of Records Formats in Messages...............26
3.5 Pull Directory Hosted on an End Station..............27
3.6 Pull Directory Message Errors........................28
3.6.1 Error Codes..29
3.6.2 Sub-Errors Under Error Codes 1 and 3...............30
3.6.3 Sub-Errors Under Error Codes 128 and 131...........30
3.7 Additional Pull Details...............................31
3.8 The No Data Flag......................................31

4. Directory Use Strategies and Push-Pull Hybrids........33
5. Security Considerations................................35

6. IANA Considerations....................................36
6.1 ESADI-Parameter Data Extensions......................36
6.2 RBridge Channel Protocol Number......................37
6.3 The Pull Directory (PUL) and No Data (NOD) Bits.......37
6.4 TRILL Pull Directory QTYPEs..........................37
6.5 Pull Directory Error Code Registries.................38

Normative References......................................39
Informational References..................................40

Acknowledgments..41
Authors' Addresses.....................................42

1. Introduction

[RFC7067] gives a problem statement and high level design for using directory servers to assist TRILL [RFC6325] edge nodes in reducing multi-destination ARP/ND [ARPreduction], reducing unknown unicast flooding traffic, and improving security against address spoofing within a TRILL campus. Because multi-destination traffic becomes an increasing burden as a network scales up in number of nodes, reducing ARP/ND and unknown unicast flooding improves TRILL network scalability. This document describes specific mechanisms for directory servers to assist TRILL edge nodes. These mechanisms are optional to implement.

The information held by the Directory(s) is address mapping and reachability information. Most commonly, what MAC address [RFC7042] corresponds to an IP address within a Data Label (VLAN or FGL (Fine Grained Label [RFC7172])) and the egress TRILL switch (RBridge), and optionally what specific TRILL switch port, from which that MAC address is reachable. But it could be what IP address corresponds to a MAC address or possibly other address mappings or reachability.

In the data center environment, it is common for orchestration software to know and control where all the IP addresses, MAC addresses, and VLANs/tenants are in a data center. Thus such orchestration software can be appropriate for providing the directory function or for supplying the Directory(s) with directory information.

Directory services can be offered in a Push or Pull Mode [RFC7067]. Push Mode, in which a directory server pushes information to TRILL switches indicating interest, is specified in Section 2. Pull Mode, in which a TRILL switch queries a server for the information it wants, is specified in Section 3. More detail on modes of operation, including hybrid Push/Pull, are provided in Section 4.

The mechanism used to initially populate directory data in primary servers is beyond the scope of this document. A primary server can use the Push Directory service to provide directory data to secondary servers as described in Section 2.5.

1.1 Uses of Directory Information

A TRILL switch can consult Directory information whenever it wants, by (1) searching through information that has been retained after being pushed to it or pulled by it or (2) by requesting information from a Pull Directory. However, the following are expected to be the most common circumstances leading to directory information use. All of these are cases of ingressing (or originating) a native frame.

1. ARP requests and replies [RFC826] are normally broadcast. But a directory assisted edge TRILL switches could intercept ARP messages and reply if the TRILL switch has the relevant information.

2. IPv6 ND (Neighbor Discovery [RFC4861]) requests and replies are normally multicast. Except in the case of Secure ND [RFC3971] where possession of the right keying material might be required, directory assisted edge TRILL switches could intercept ND messages and reply if the TRILL switch has the relevant information.

3. Unknown destination MAC addresses. An edge TRILL switch ingressing a native frame necessarily has to determine if it knows the egress RBridge from which the destination MAC address of the frame (in the frame's VLAN or FGL) is reachable. It might learn that information from the directory or could query the directory if it does not know. Furthermore, if the edge TRILL switch has complete directory information, it can detect a forged source MAC address in the native frame and discard the frame in that case.

4. RARP [RFC903] is similar to ARP as above.

1.2 Terminology

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].

The terminology and acronyms of [RFC6325] are used herein along with the following:

CSNP Time: Complete Sequence Number PDU Time. See ESDADI [RFC7357] and Section 6.1 below.

Data Label: VLAN or FGL.

FGL: Fine Grained Label [RFC7172].

Host: Application running on a physical server or a virtual machine. A host must have a MAC address and usually has at least one IP address.

IP: Internet Protocol. In this document, IP includes both IPv4 and IPv6.

MacDA: Destination MAC address.

PDSS: Push Directory Server Status. See Sections 2 and 6.1 below.

PUL: Pull Directory flag bit. See Sections 3 and 6.3 below.

primary server: A Directory server that obtains the information it is serving up by a reliable mechanism outside the scope of this document designed to assure the freshness of that information. (See secondary server.)

RBridge: An alternative name for a TRILL switch.

secondary server: A Directory server that obtains the information it is serving up from one or more primary servers.

TRILL: Transparent Interconnection of Lots of Links or Tunneled Routing in the Link Layer.

TRILL switch: A device that implements the TRILL protocol.

2. Push Model Directory Assistance Mechanisms

In the Push Model [RFC7067], one or more Push Directory servers reside at TRILL switches and push down the address mapping information for the various addresses associated with end station interfaces and the TRILL switches from which those interfaces are reachable [IA]. This service is scoped by Data Label (VLAN or FGL [RFC7172]). A Push Directory also advertises whether or not it believes it has pushed complete mapping information for a Data Label. It might be pushing only a subset of the mapping and/or reachability information for a Data Label. The Push Model uses the ESADI [RFC7357] protocol as its distribution mechanism.

With the Push Model, if complete address mapping information for a Data Label is being pushed, a TRILL switch (RBridge) which has that complete information and is ingressing a native frame can simply drop the frame if the destination unicast MAC address can't be found in the mapping information available, instead of flooding the frame (ingressing it as an unknown MAC destination TRILL Data frame). But this will result in lost traffic if ingress TRILL switch's directory information is incomplete.

2.1 Requesting Push Service

In the Push Model, it is necessary to have a way for a TRILL switch to subscribe to information from the directory server(s). TRILL switches simply use the ESADI [RFC7357] protocol mechanism to announce, in their core IS-IS LSPs, the Data Labels for which they are participating in ESADI by using the Interested VLANs and/or Interested Labels sub-TLVs [RFC7176]. This will cause them to be pushed the Directory information for all such Data Labels that are being served by the one or more Push Directory servers.

2.2 Push Directory Servers

Push Directory servers advertise their availability to push the mapping information for a particular Data Label to each other and to ESADI participants for that Data Label through ESADI by setting the PDSS (Push Directory Server Status) in their ESADI Parameter APPsub-TLV for that ESADI instance (see [RFC7357] and Section 6.1) to a non-zero value. Each Push Directory server MUST participate in ESADI for the Data Labels for which it will push mappings and set the PDSS field in its ESADI-Parameters APPsub-TLV for that Data Label.

For robustness, it is useful to have multiple Push Directory Servers for each Data Label. Each Push Directory server is configured with a number N in the range 1 to 8, which defaults to 2, for each Data Label for which it can push directory information. If the Push Directory servers for a Data Label are configured consistently with the same N and at least N servers are available, then N copies of that directory will will be pushed.

Each Push Directory server also has an 8-bit priority to be Active (see Section 6.1 of this document). This priority is treated as an unsigned integer where larger magnitude means higher priority. This priority appears in its ESADI Parameter APPsub-TLV.

For each Data Label it can serve, each Push Directory server checks to see if there are enough higher priority servers to push the desired number of copies. It does this by ordering, by priority, the Push Directory servers that it can see in the ESADI link state database for that Data Label that are data reachable [rfc7180bis] and determines its own position in that order. If a Push Directory server is configured to believe that N copies of the mappings for a Data Label should be pushed and finds that it is number K in the priority ordering (where the first is highest priority and the last is lowest), then if K is less than or equal to N the Push Directory server is Active. If K is greater than N it is Stand-By. Active and Stand-By behavior are specified below.

For a Push Directory to reside on an end station, one or more TRILL switches locally connected to that end station must proxy for the Push Directory server and advertise themselves as Push Directory servers. It appears to the rest of the TRILL campus that these TRILL switches (that are proxying for the end station) are the Push Directory server(s). The protocol between such a Push Directory end station and the one or more proxying TRILL switches acting as Push Directory servers is beyond the scope of this document.

2.3 Push Directory Server State Machine

The subsections below describe the states, events, and corresponding actions for Push Directory servers.

The meaning of the value of the PDSS field in a Push Directory's ESADI Parameter APPsub-TLV is summarized in the table below.

PDSS	Meaning
0	Not a Push Directory Server
1	Push Directory Server in Stand-By Mode
2	Push Directory Server in Active Mode but not complete
3	Push Directory Server in Active Mode that has pushed

```
                  complete data
```

2.3.1 Push Directory States

A Push Directory Server is in one of seven states, as listed below, for each Data Label it can serve. The name of each state is followed by a symbol that starts and ends with an angel bracket and represents the state. The value that the Push Directory Server advertises in PDSS is determined by the state. In addition, it has an internal State-Transition-Time variable for each Data Label it serves which is set at each state transition and which enables it to determine how long it has been in its current state for that Data Label.

Down <S1>: A completely shut down virtual state defined for convenience in specifying state diagrams. A Push Directory Server in this state does not advertise any Push Directory data. It may be participating in ESDADI [RFC7357] with the PDSS field zero in its ESADI-Parameters or might be not participating in ESADI at all. (All states other than the Down state are considered to be Up states and imply a non-zero PDSS field.)

Stand-By <S2>: No Push Directory data is advertised. Any outstanding EASDI-LSP fragments containing directory data are updated to remove that data and if the result is an empty fragment (contains nothing except possibly an Authentication TLV), the fragment is purged. The Push Directory participates in ESDADI [RFC7357] and advertises its ESADI fragment zero that includes an ESADI-Parameters APPsub-TLV with the PDSS field set to 1.

Active <S3>: The PDSS field in the ESADI-Parameters is set to 2. If a Push Directory server is Active, it advertises its directory data and any changes through ESADI [RFC7357] in its ESADI-LSPs using the Interface Addresses [IA] APPsub-TLV and updates that information as it changes.

Active Completing <S4>: Same behavior as the Active state except that it responds differently to events. The purpose of this state is to be sure there has been enough time for directory information to propagate to subscribing edge TRILL switches before the Directory Server advertises that the information is complete.

Active Complete <S5>: The same behavior as Active except that the PDSS field in the ESADI-Parameters APPsub-TLV is set to 3 and the server responds differently to events.

Going Stand-By <S6>: The same behavior as Active except that it responds differently to events. The purpose of this state is to be sure that the information, that the directory is no longer complete, has enough time to propagate to edge TRILL switches before the Directory Server stops advertising updates to the information.

Active Uncompleting <S7>: The same behavior as Active except that it responds differently to events. The purpose of this state is to be sure that the information, that the directory is no longer complete, has enough time to propagate to edge TRILL switches before the Directory Server might stop advertising updates to the information. (See note below.)

Note: It might appear that a Push Directory could transition directly from Active Complete to Active, since Active state continues to advertise updates, eliminating the need for the Active Uncompleting transition state. But consider the case of the Push Directory being configured to be incomplete and then the Stand-By Condition (see Section 2.3.2) occurring immediately thereafter. If the first of these two events caused the server to transition directly to the Active state then, when the Stand-By Condition occurred, it would immediately transition to Stand-By and stop advertising updates even though there might not have been enough time for knowledge of its incompleteness to have propagated to all edge TRILL switches.

The following table summarizes PDSS value for each state:

State	PDSS
Down <S1>	0
Stand-By <S2>	1
Active <S3>	2
Active Completing <S4>	2
Active Complete <S5>	3
Going Stand-By <S6>	2
Active Uncompleting <S7>	2

2.3.2 Push Directory Events and Conditions

Three auxiliary conditions referenced later in this section are defined as follows for convenience:

The Activate Condition: In order to have the desired number of Push Directory servers pushing data, this Push Directory server should be active. This is determined by the server finding that it is priority K among the data reachable Push Directory servers (where highest priority is 1), it is configured that there should be N copies pushed, and K is less than or equal to N. For example, the Push Directory server is configured that 2 copies should be pushed and finds that it is priority 1 or 2 among the Push Directory servers it can see.

The Stand-By Condition: In order to have the desired number of Push

Directory servers pushing data, this Push Directory server should be stand-by (not active). This is determined by the server finding that it is priority K among the data reachable Push Directory servers (where highest priority is 1), it is configured that there should be N copies pushed, and K is greater than N. For example, the Push Directory server is configured that 2 copies should be pushed and finds that it is priority 3 or lower priority (higher number) among the Push directory servers it can see.

The Time Condition: The Push Directory server has been in its current state for a configurable amount of time that defaults to twice its CSNP time (see Section 6.1).)

The events and conditions listed below cause state transitions in Push Directory servers.

1. Push Directory server was Down but is now Up.

2. The Push Directory server or the TRILL switch on which it resides is being shut down.

3. The Activate Condition is met and the server is not configured to believe it has complete data.

4. The Stand-By Condition is met.

5. The Activate Condition is met and the server is configured to believe it has complete data.

6. The server is configured to believe it does not have complete data.

7. The Time Condition is met.

2.3.3 State Transition Diagram and Table

The state transition table is as follows:

State	Down	Stand-By	Active	Active	Active	Going	Active
				Completing	Complete	Stand-By	Uncompleting
Event	<S1>	<S2>	<S3>	<S4>	<S5>	<S6>	<S7>
1	<S2>	<S2>	<S3>	<S4>	<S5>	<S6>	<S7>
2	<S1>	<S1>	<S2>	<S2>	<S6>	<S6>	<S7>
3	<S1>	<S3>	<S3>	<S3>	<S7>	<S3>	<S7>
4	<S1>	<S2>	<S2>	<S2>	<S6>	<S6>	<S6>
5	<S1>	<S4>	<S4>	<S4>	<S5>	<S5>	<S5>

```
     6  |<S1>|  <S2>  | <S3> |   <S3>   |  <S7>  |  <S6>  |    <S7>
     7  |<S1>|  <S2>  | <S3> |   <S5>   |  <S5>  |  <S2>  |    <S3>
```

The above state table is equivalent to the following transition diagram:

```
     +-----------+
     | Down <S1> |<---------+
     +-----------+          |
       |1  ^   | 3,4,5,6,7  |
       |   |   +------------+
       V   |2
     +---------------+
     | Stand-By <S2> |<------------------------------------------+
     +---------------+    ^   ^            ^                     |
       |5   |3  |1,4,6,7  |   |            |                     |
       |    |   +---------+   |            |                     |
       |    V                 |2,4         |                     |
       |  +---------------------+          |                     |
       |  | Active <S3>          |<---------|-----------------+   |
       |  +---------------------+     ^    |                 |   |
       |   |5  ^    |1,3,6,7  ^       |    |                 |   |
       |   |   |    |         |       |    |                 |   |
       |   |   |    +---------+       |    |                 |   |
       |   |   |                      |    |                 |   |
       V   V   |3,6                   |    |                 |   |
     +-------------------------+       |    |                 |   |
     | Active Completing <S4> |------------+                 |   |
     +------------------------+ 2,4   |                     |   |
       |7  |1,5    ^                  |                     |   |
       |   |       |                  |                     |   |
       |   +-------+                  |                     |   |
       |                              |                     |   |
       |          +--------------------------------------+   |   |
       |          |                   |                  |   |   |
       V          V                   |7                 |5  |3  |7
     +---------------+ 3,6    +----------------+ 4    +----------------+
     |    Active     |------->|     Active     |----->|     Going      |
     |   Complete    |        |  Uncompleting  |      |    Stand-By    |
     |     <S5>      |<-------|      <S7>      |      |      <S6>      |
     +---------------+      5 +----------------+      +----------------+
       |1,5,7  ^   |2,4         |1,2,3,6     ^          ^   |1,2,4,6 ^
       |       |   |            |            |          |   |        |
       +-------+   |            +------------+          |   +--------+
                   |                                    |
                   +------------------------------------+
```

Figure 2. Push Server State Diagram

2.4 Additional Push Details

Push Directory mappings can be distinguished from other data distributed through ESADI because mappings are distributed only with the Interface Addresses APPsub-TLV [IA] and are flagged in that APPsub-TLV as being Push Directory data.

TRILL switches, whether or not they are a Push Directory server, MAY continue to advertise any locally learned MAC attachment information in ESADI [RFC7357] using the Reachable MAC Addresses TLV [RFC6165]. However, if a Data Label is being served by complete Push Directory servers, advertising such locally learned MAC attachment generally SHOULD NOT be done as it would not add anything and would just waste bandwidth and ESADI link state space. An exception might be when a TRILL switch learns local MAC connectivity and that information appears to be missing from the directory mapping.

Because a Push Directory server needs to advertise interest in one or more Data Labels even though it might not want to receive multi-destination data in those Data Labels, the No Data (NOD) flag bit is provided as discussed in Section 3.8.

When a Push Directory server is no longer data reachable [rfc7180bis], TRILL switches MUST ignore any Push Directory data from that server because it is no longer being updated and may be stale.

The nature of dynamic distributed asynchronous systems is such that it is impossible for a TRILL switch receiving Push Directory information to be absolutely certain that it has complete information. However, it can obtain a reasonable assurance of complete information by requiring two conditions to be met:

1. The PDSS field is 3 in the ESADI zero fragment from the server for the relevant Data Label.
2. In so far as it can tell, it has had continuous data connectivity to the server for a configurable amount of time that defaults to twice the server's CSNP time.

Condition 2 is necessary because a client TRILL switch might be just coming up and receive an EASDI LSP meeting the requirement in condition 1 above but has not yet received all of the ESADI LSP fragment from the Push Directory server.

There may be conflicts between mapping information from different Push Directory servers or conflicts between locally learned information and information received from a Push Directory server. In case of such conflicts, information with a higher confidence value [RFC6325] is preferred over information with a lower confidence. In case of equal confidence, Push Directory information is preferred to locally learned information and if information from Push Directory servers conflicts, the information from the higher priority Push Directory server is preferred.

2.5 Primary to Secondary Server Push Service

A secondary Push or Pull Directory server is one that obtains its data from a primary directory server. Other techniques MAY be used but, by default, this data transfer occurs through the primary server acting as a Push Directory server for the Data Labels involved while the secondary directory server takes the pushed data it receives from the highest priority Push Directory server and re-originates it. Such a secondary server may be a Push Directory server or a Pull Directory server or both for any particular Data Label. Because the data from a secondary server will necessarily be at least a little less fresh than that from a primary server, it is RECOMMENDED that the re-originated secondary server data be given a confidence level of one less than that of the data as received from the primary (or unchanged if it is already of minimum confidence).

3. Pull Model Directory Assistance Mechanisms

In the Pull Model [RFC7067], a TRILL switch (RBridge) pulls directory information from an appropriate Directory Server when needed.

Pull Directory servers for a particular Data Label X are found by looking in the core TRILL IS-IS link state database for data reachable [rfc7180bis] TRILL switches that advertise themselves by having the Pull Directory flag (PUL) on in their Interested VLANs or Interested Labels sub-TLV (see Section 6.3)) for that Data Label. If multiple such TRILL switches indicate that they are Pull Directory Servers for a particular Data Label, pull requests can be sent to any one or more of them but it is RECOMMENDED that pull requests be preferentially sent to the server or servers that are lowest cost from the requesting TRILL switch.

Pull Directory requests are sent by enclosing them in an RBridge Channel [RFC7178] message using the Pull Directory channel protocol number (see Section 6.2). Responses are returned in an RBridge Channel message using the same channel protocol number. See Section 3.2 for Query and Response Message formats. For cache consistency or notification purposes, Pull Directory servers, under certain conditions, MUST send unsolicited Update Messages to client TRILL switches they believe may be holding old data and those clients can acknowledge such updates, as described in Section 3.3. All these messages have a common header as described in Section 3.1. Errors can be returned for queries or updates as described in Section 3.6.

The requests to Pull Directory Servers are typically derived from ingressed ARP [RFC826], ND [RFC4861], or RARP [RFC903] messages, or data frames with unknown unicast destination MAC addresses, intercepted by an ingress TRILL switch as described in Section 1.1.

Pull Directory responses include an amount of time for which the response should be considered valid. This includes negative responses that indicate no data is available. It is RECOMMENDED that both positive responses with data and negative responses can be cached and used to locally handle ARP, ND, RARP, unknown destination MAC frames, or the like, until the responses expire. If information previously pulled is about to expire, a TRILL switch MAY try to refresh it by issuing a new pull request but, to avoid unnecessary requests, SHOULD NOT do so if it has not been recently used. The validity timer of cached Pull Directory responses is NOT reset or extended merely because that cache entry is used.

3.1 Pull Directory Message Common Format

All Pull Directory messages are transmitted as the payload of RBridge Channel messages [RFC7178]. Pull Directory messages are formatted as described herein starting with the following common 8-byte header:

```
                       1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 3 3
   0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |  Ver  | Type  | Flags | Count |      Err      |    SubErr     |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |                        Sequence Number                        |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  | Type Specific Payload - variable length
  +-+-+- ...
```

Ver: Version of the Pull Directory protocol as an unsigned integer. Version zero is specified in this document.

Type: The Pull Directory message type as follows:

Type	Section	Name
0	-	Reserved
1	3.2.1	Query
2	3.2.2	Response
3	3.3.1	Update
4	3.3.2	Acknowledge
5-14	-	Unassigned
15	-	Reserved

Flags: Four flag bits whose meaning depends on the Pull Directory message Type. Flags whose meanings are not specified are reserved, MUST be sent as zero, and MUST be ignored on receipt.

Count: Pull Directory message types specified herein have zero or more occurrences of a Record as part of the type specific payload. The Count field is the number of occurrences of that Record as an unsigned integer. For any Pull Directory messages not structured with such occurrences, this field MUST be sent as zero and ignored on receipt.

Err, SubErr: The error and suberror fields are only used in messages that are in the nature of replies. In messages that are requests or updates, these fields MUST be sent as zero and ignored on receipt. An Err field containing the value zero means no error. The meaning of values in the SubErr field depends on the value of the Err field but in all cases, a zero SubErr field is allowed and provides no additional information beyond the value of the Err field.

Sequence Number: An identifying 32-bit quantity set by the TRILL switch sending a request or other unsolicited message and returned in every corresponding reply or acknowledgement. It is used to match up responses with the message to which they respond.

Type Specific Payload: Format depends on the Pull Directory message Type.

3.2 Pull Directory Query and Response Messages

The format of the Pull Directory Query and Response Messages is specified below.

3.2.1 Pull Directory Query Message Format

A Pull Directory Query Message is sent as the Channel Protocol specific content of an RBridge Channel message [RFC7178] TRILL Data packet or as a native RBridge Channel data frame (see Section 3.5). The Data Label of the packet is the Data Label in which the query is being made. The priority of the channel message is a mapping of the priority of the frame being ingressed that caused the query with the default mapping depending, per Data Label, on the strategy (see Section 4) or a configured priority for generated queries. (Generated queries are those not the result of a mapping. For example, a query to refresh a cache entry.) The Channel Protocol specific data is formatted as a header and a sequence of zero or more QUERY Records as follows:

```
                       1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 3 3
   0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |  Ver  | Type  | Flags | Count |      Err      |    SubErr     |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |                        Sequence Number                        |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  | QUERY 1
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-...
  | QUERY 2
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-...
  | ...
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-...
  | QUERY K
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-...
```

Ver, Sequence Number: See 3.1.

Type: 1 for Query. Queries received by an TRILL switch that is not a Pull Directory for the relevant Data Label result in an error response (see Section 3.6) unless inhibited by rate limiting. (See [RFC7178] for response if the Pull Directory RBridge Channel protocol is not enabled.)

Flags, Err, and SubErr: MUST be sent as zero and ignored on receipt.

Count: Number of QUERY Records present. A Query Message Count of zero is explicitly allowed, for the purpose of pinging a Pull Directory server to see if it is responding. On receipt of such an empty Query Message, a Response Message that also has a Count of zero is sent unless inhibited by rate limiting.

QUERY: Each QUERY Record within a Pull Directory Query Message is formatted as follows:

```
       0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
     +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
     |        SIZE           |FL|  RESV  |   QTYPE   |
     +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
   If QTYPE = 1
     +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
     |                      AFN                      |
     +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
     |  Query address ...
     +--+--+--+--+--+--+--+--+--+--+--+--...
   If QTYPE = 2, 3, 4, or 5
     +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
     |  Query frame ...
     +--+--+--+--+--+--+--+--+--+--+--+--...
```

SIZE: Size of the QUERY Record in bytes as an unsigned integer not including the SIZE field and following byte. A value of SIZE so large that the material doesn't fit in the Query Message indicates a malformed QUERY Record. The QUERY Record with the illegal SIZE value and any subsequent QUERY Records MUST be ignored and the entire Query Message MAY be ignored.

FL: The FLooded flag that is ignored if QTYPE is zero. If QTYPE is 2 through 5 and the directory information sought is not found, the frame provided is flooded, otherwise it is not forwarded. See Section 3.2.2.2.

RESV: A block of three reserved bits. MUST be sent as zero and ignored on receipt.

QTYPE: There are several types of QUERY Records currently defined in two classes as follows: (1) a QUERY Record that provides an explicit address and asks for all addresses for the interface specified by the query address and (2) a QUERY Record that includes a frame. The fields of each are specified below. Values of QTYPE are as follows:

QTYPE	Description
0	Reserved
1	Address query
2	ARP query frame
3	ND query frame
4	RARP query frame
5	Unknown unicast MAC query frame
6-14	Unassigned
15	Reserved

AFN: Address Family Number of the query address.

Query Address: The query is asking for any other addresses, and the nickname of the TRILL switch from which they are reachable, that correspond to the same interface, within the data label of the query. Typically that would be either (1) a MAC address with the querying TRILL switch primarily interested in the TRILL switch by which that MAC address is reachable, or (2) an IP address with the querying TRILL switch interested in the corresponding MAC address and the TRILL switch by which that MAC address is reachable. But it could be some other address type.

Query Frame: Where a QUERY Record is the result of an ARP, ND, RARP, or unknown unicast MAC destination address, the ingress TRILL switch MAY send the frame to a Pull Directory Server if the frame is small enough that the resulting Query Message fits into a TRILL Data packet within the campus MTU.

If no response is received to a Pull Directory Query Message within a timeout configurable in milliseconds that defaults to 100, the Query Message should be re-transmitted with the same Sequence Number up to a configurable number of times that defaults to three. If there are multiple QUERY Records in a Query Message, responses can be received to various subsets of these QUERY Records before the timeout. In that case, the remaining unanswered QUERY Records should be re-sent in a new Query Message with a new sequence number. If a TRILL switch is not capable of handling partial responses to queries with multiple QUERY Records, it MUST NOT send a Request Message with more than one QUERY Record in it.

See Section 3.6 for a discussion of how Query Message errors are handled.

3.2.2 Pull Directory Responses

A Pull Directory Query Message results in a Pull Directory Response Message as described in Section 3.2.2.1.

In addition, if the QUERY Record QTYPE was 2, 3, 4, or 5, the frame included in the Query may be modified and forwarded by the Pull Directory server as described in Section 3.2.2.2.

3.2.2.1 Pull Directory Response Message Format

Pull Directory Response Messages are sent as the Channel Protocol specific content of an RBridge Channel message [RFC7178] TRILL Data packet or as a native RBridge Channel data frame (see Section 3.5). Responses are sent with the same Data Label and priority as the Query Message to which they correspond except that the Response Message priority is limited to be not more than a configured value. This priority limit is configurable per TRILL switch and defaults to priority 6. Pull Directory Response Messages SHOULD NOT be sent with priority 7 as that priority SHOULD be reserved for messages critical to network connectivity.

The RBridge Channel protocol specific data format is as follows:

```
                        1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 3 3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |  Ver  | Type  | Flags | Count |      Err      |    SubErr     |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                        Sequence Number                        |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   | RESPONSE 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-...
   | RESPONSE 2
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-...
   | ...
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-...
   | RESPONSE K
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-...
```

Ver, Sequence Number: As specified in Section 3.1.

Type: 2 = Response.

Flags: MUST be sent as zero and ignored on receipt.

Count: Count is the number of RESPONSE Records present in the

Response Message.

Err, SubErr: A two-part error code. Zero unless there was an error in the Query Message, for which case see Section 3.6.

RESPONSE: Each RESPONSE Record within a Pull Directory Response Message is formatted as follows:

```
   0   1   2   3   4   5   6   7   8   9 10  11 12 13 14 15
 +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
 |         SIZE          |OV|  RESV  |   Index   |
 +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
 |                   Lifetime                    |
 +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
 |                Response Data ...
 +--+--+--+--+--+--+--+--+--+--+--+--...
```

SIZE: The size of the RESPONSE Record is an unsigned integer number of bytes not including the SIZE field and following byte. A value of SIZE so large that the material doesn't fit in the Query Message indicates a malformed QUERY Record. The QUERY Record with such an excessive SIZE value and any subsequent QUERY Records MUST be ignored and the entire Query Message MAY be ignored.

OV: The overflow flag. Indicates, as described below, that there was too much Response Data to include in one Response Message.

RESV: Three reserved bits that MUST be sent as zero and ignored on receipt.

Index: The relative index of the QUERY Record in the Query Message to which this RESPONSE Record corresponds. The index will always be one for Query Messages containing a single QUERY Record. If the Index is larger than the Count was in the corresponding Query, that RESPONSE Record MUST be ignored and subsequent RESPONSE Records or the entire Response Message MAY be ignored.

Lifetime: The length of time for which the response should be considered valid in units of 100 milliseconds except that the values zero and 2**16-1 are special. If zero, the response can only be used for the particular query from which it resulted and MUST NOT be cached. If 2**16-1, the response MAY be kept indefinitely but not after the Pull Directory server goes down or becomes unreachable. (The maximum definite time that can be expressed is a little over 1.8 hours.)

Response Data: There are three types of RESPONSE Records.

- If the Err field of the enclosing Respose Message has a message level error code in it, then the the REPONSE Records are omitted and Count will be zero. See Section 3.6 for additional information on errors.

- If the Err field of the encosing Response Message has a record level error code in it, then the RESPONSE Records are those in error as further described in Section 3.6.
- If the Err field of the enclosing Repose Message is zero, then the Response Data in each RESPONSE Record is formatted as the value of an Interface Addresses APPsub-TLV [IA]. The maximum size of such contents is 255 bytes in the case when the RESPONSE Record SIZE field is 255.

Multiple RESPONSE Records can appear in a Response Message with the same Index if the answer to a QUERY Record consists of multiple Interface Address APPsub-TLV values. This would be necessary if, for example, a MAC address within a Data Label appears to be reachable by multiple TRILL switches. However, all RESPONSE Records to any particular QUERY Record MUST occur in the same Response Message. If a Pull Directory holds more mappings for a queried address than will fit into one Response Message, it selects which to include by some method outside the scope of this document and sets the overflow flag (OV) in all of the RESPONSE Records responding to that query address.

See Section 3.6 for a discussion of how errors are handled.

3.2.2.2 Pull Directory Forwarding

Query Messages with QTYPEs 2, 3, 4, and 5 are interpreted and handled as described below. In these cases, if the information sought is not in the directory, the provided frame is forwarded by the Pull Directory server as a multi-destination TRILL Data packet if the FL flag in the Query Message was one, otherwise the frame is not forwarded. If there was no error in the handling of the enclosing Query Message then the Pull Directory server forwards the frame inside that QUERY Record, after modifying it in some cases, as described below.

ARP: When QTYPE is 2, an ARP [RFC826] frame is included in the QUERY Record. The ar$op field MUST be ares_op$REQUEST and for the response described in 3.2.2.1, this is treated as a query for the target protocol address where the AFN of that address is given by ar$pro. (ARP field and value names with embedded dollar signs are specified in [RFC826].) If ar$op is not ares_op$REQUEST or the ARP is malformed or the query fails, an error is returned. Otherwise the ARP is modified into the appropriate ARP response that is then sent by the Pull Directory server as a TRILL Data packet.

ND: When QTYPE is 3, an IPv6 Neighbor Discover (ND [RFC4861]) frame is included in the QUERY Record. Only Neighbor Solicitation ND frames (corresponding to an ARP query) are allowed. An error is returned for other ND frames or if the target address is not found. Otherwise an ND Neighbor Advertisement response is returned by the Pull Directory server as a TRILL Data packet.

RARP: When QTYPE is 4, a RARP [RFC903] frame is included in the QUERY Record. If the ar$op field is ares_op$REQUEST, the frame is handled as an ARP as described above. Otherwise the ar$op field MUST be 'reverse request' and for the response described in 3.2.2.1, this is treated as a query for the target hardware address where the AFN of that address is given by ar$hrd. (See [RFC826] for RARP fields.) If ar$op is not one of these values or the RARP is malformed or the query fails, an error is returned. Otherwise the RARP is modified into the appropriate RARP response that is then unicast by the Pull Directory server as a TRILL Data packet to the source hardware MAC address.

MacDA: When QTYPE is 5, indicating a fame is provided in the QUERY Record whose destination MAC address TRILL switch attachment is unknown, the only requirement is that this MAC address must be unicast. If it is group addressed an error is returned. For the response described in 3.2.2.1, it is treated as a query for the MacDA. If the Pull Directory contains TRILL switch attachment information for the MAC address in the Data Label of the Query Message, it forwards the frame to that switch in a unicast TRILL Data packet.

3.3 Cache Consistency

Unless it sends all responses with a Lifetime of zero, a Pull Directory MUST take action, by sending Update Messages, to minimize the amount of time that a TRILL switch will continue to use stale information from that Pull Directory. The format of Update Messages and the Acknowledge Messages used to respond to Update Messages are given in Sections 3.3.1 and 3.3.2.

A Pull Directory server MUST maintain one of the following three sets of records, in order of increasing specificity. Retaining more specific records, such as that given in method 3 below, minimizes spontaneous Update Messages sent to update pull client TRILL switch caches but increases the record keeping burden on the Pull Directory server. Retaining less specific records, such as that given in method 1, will generally increase the volume and overhead due to spontaneous Update Messages and due to unnecessarily invalidating cached information, but will still maintain consistency and will reduce the record keeping burden on the Pull Directory server. In all cases, there may still be brief periods of time when directory information has changed, but information a pull client has cached has not yet been updated or expunged.

1. An overall record per Data Label of when the last positive response data sent will expire at some requester and when the last negative response will expire at some requester, assuming those requesters cached the response.

2. For each unit of data (IA APPsub-TLV Address Set [IA]) held by the server and each address about which a negative response was sent, when the last response sent with that positive response data and when the last negative response will expire at a requester, assuming the requester cached the response.

3. For each unit of data held by the server (IA APPsub-TLV Address Set [IA]) and each address about which a negative response was sent, a list of TRILL switches that were sent that data as a positive response or sent a negative response for the address, and the expected time to expiration for that data or address at each such TRILL switch, assuming the requester cached the response.

RESPONSE Records sent with a zero lifetime are considered to have already expired and so do not need to be tracked.

A Pull Directory server may have a limit as to how many TRILL switches for which it can maintain expiry information by method 3 above or how many data units or addresses it can maintain expiry information for by method 2 or the like. If such limits are exceeded, it MUST transition to a lower numbered method but, in all cases, MUST support, at a minimum, method 1.

When data at a Pull Directory is changed, deleted, or added and there may be unexpired stale information at a requesting TRILL switch, the Pull Directory MUST send an Update Message as discussed below. The sending of such an Update Message MAY be delayed by a configurable number of milliseconds that default to 50 milliseconds to await other possible changes that could be included in the same Update.

1. If method 1, the crudest method, is being followed, then when any Pull Directory information in a Data Label is changed or deleted and there are outstanding cached positive data response(s), an all-addresses flush positive data Update Message is flooded within that Data Label as an RBridge Channel Message with an Inner.MacDA of All-Egress-RBridges. Similarly if data is added and there are outstanding cached negative responses, an all-addresses flush negative message is similarly flooded. The Count field being zero in an Update Message indicates "all-

addresses". On receiving an all-addresses flooded flush positive Update from a Pull Directory server it has used, indicated by the F and P bits being one and the Count being zero, a TRILL switch discards the cached data responses it has for that Data Label. Similarly, on receiving an all addresses flush negative Update, indicated by the F and N bits being one and the Count being zero, it discards all cached negative replies for that Data Label. A combined flush positive and negative can be flooded by having all of the F, P, and N bits set to one resulting in the discard of all positive and negative cached information for the Data Label.

2. If method 2 is being followed, then a TRILL switch floods address specific positive Update Messages when data that might be cached by a querying TRILL switch is changed or deleted and floods address specific negative Update Messages when such information is added to. Such messages are somewhat similar to the method 1 flooded flush Update Messages and are also sent as RBridge Channel messages with an Inner.MacDA of All-Egress-RBridges. However the Count field will be non-zero and either the P or N bit, but not both, will be one. There are actually four possible message types that can be flooded:

   2.a If data still being cached is updated, then an Update Message is sent with the P flag set and the Err field zero. The addresses in the RESPONSE Records in the unsolicited response are compared to the addresses about which the receiving TRILL switch is holding cached positive information from that server and, if they match, the cached information is updated.

   2.b If data still being cached is deleted, then an Update Message is sent with the P flag set and the Err field non-zero giving the error that would now be encountered in attempting to pull information for the relevant address from the Pull Directory server. In this non-zero Err field case, the RESPONSE Record(s) differ from non-zero Err Reply Message RESPONSE Records in that they include an interface address set. Any cached positive information for the address is deleted and the negative response cached as per the lifetime given.

   2.c If data for an address about which a negative response was sent is added so that negative response is now incorrect, an Update Message is sent with the N flag set to one and the Err field zero. The addresses in the RESPONSE Records in the unsolicited response are compared to the addresses about which the receiving TRILL switch is holding cached negative information from that server and, if they match, the cached negative information is deleted and the positive information provided is cached as per the lifetime given.

2.d In the rare case where it is desired to change the lifetime or error associated with cached negative information, it is possible to send an Update Message with the N flag set to one and the Err field non-zero. As in case 2.b above, the RESPONSE Record(s) give the relevant addresses. Any cached negative information for the address is updated.

3. If method 3 is being followed, the same sort of unsolicited Update Messages are sent as with method 2 above except they are not normally flooded but unicast only to the specific TRILL switches the directory server believes may be holding the cached positive or negative information that needs updating. However, a Pull Directory server MAY flood unsolicited updates under method 3, for example if it determines that a sufficiently large fraction of the TRILL switches in some Data Label are requesters that need to be updated.

A Pull Directory server tracking cached information with method 3 MUST NOT clear the indication that it needs to update cached information at a querying TRILL switch until it has sent an Update Message and received a corresponding Acknowledge Message or it has sent a configurable number of updates at a configurable interval which default to 3 updates 100 milliseconds apart.

A Pull Directory server tracking cached information with methods 2 or 1 SHOULD NOT clear the indication that it needs to update cached information until it has sent an Update Message and received a corresponding Acknowledge Message from all of its ESADI neighbors or it has sent a configurable number of updates at a configurable interval that defaults to 3 updates 100 milliseconds apart.

3.3.1 Update Message Format

An Update Message is formatted as a Response Message with the differences described in Section 3.3 above and the following:

- The Type field in the message header is set to 3.
- The Err field in the message header MUST be sent as zero and ignored on receipt.
- The Index field in the RESPONSE Record(s) is set to zero (but the Count field in the Update Message header MUST still correctly indicate the number of RESPONSE Records present).

Update Messages are initiated by a Pull Directory server. The Sequence number space used is controlled by the originating Pull Directory server and different from Sequence number space used in a Query and the corresponding Response that are controlled by the querying TRILL switch.

The 4-bit Flags field of the message header for an Update Message is as follows:

```
      +---+---+---+---+
      | F | P | N | R |
      +---+---+---+---+
```

F: The Flood bit. If zero, the Update Message is unicast. If F=1, it is multicast to All-Egress-RBridges.

P, N: Flags used to indicate positive or negative Update Messages. P=1 indicates positive. N=1 indicates negative. Both may be 1 for a flooded all addresses Update.

R: Reserved. MUST be sent as zero and ignored on receipt

For tracking methods 2 and 3 in Section 3.3.1, a particular Update Message must have either the P flag or the N flag set but not both.

3.3.2 Acknowledge Message Format

An Acknowledge Message is sent in response to an Update Message to confirm receipt or indicate an error, unless response is inhibited by rate limiting. It is also formatted as a Response Message but the Type is set to 4.

If there are no errors in the processing of an Update Message or if there is a message level overall or header error in an Update Message, the message is essentially echoed back with the Err and SubErr fields set approriately, the Type changed to Acknowledge, and a null records section with the Count field set to zero.

If there is a record level error in an Update Message, one or more Acknowledge Messages may be returned with the erroneous record(s) indicated in Section 3.5.

3.4 Summary of Records Formats in Messages

As specified in Section 3.2 and 3.3, the Query, Response, Update, and Acknowledge Messages can have zero or more repeating Record structures under different circumstances, as summarized below. The "Err" column abbreivations in this table have the meanings listed below. "IA APPsubTLV value" means the value part of the IA APPsub-TLV specified in [IA].

MBZ = MUST be zero
Z   = zero
NZ = non-zero
NZM = non-zero message level error
NZR = non-zero record level error

Message	Err	Section	Record Structure	Response Data
Query	MBZ	3.2.1	QUERY Record	-
Response	Z	3.2.2.1	RESPONSE Record	IA APPsubTLV value
Response	NZM	3.2.2.1	null	-
Response	NZR	3.2.2.1	RESPONSE Record	Records with error
Update	MBZ	3.3.1	RESPONSE Record	IA APPsubTLV value
Acknowledge	Z	3.3.2	null	-
Acknowledge	NZM	3.3.2	null	-
Acknowledge	NZR	3.3.2	RESPONSE Record	Records with error

See Section 3.6 for further details on errors.

3.5 Pull Directory Hosted on an End Station

Optionally, a Pull Directory actually hosted on an end station MAY be supported. In that case, one or more TRILL switches must proxy for the end station and advertise themselves as Pull Directory servers. Such proxies must have a direct connection to the end station, that is a connection not involving any intermediate TRILL switches.

When the proxy Pull Directory server TRILL switch receives a Query Message, it modifies the inter-RBridge Channel message received into a native RBridge Channel message and forwards it to the end station Pull Directory server. Later, when it receives one or more responses from that end station by native RBridge Channel messages, it modifies them into inter-RBridge Channel messages and forwards them to the source TRILL switch of the original Query Message. Similarly, an Update from the end station is forwarded to client TRILL switches and acknowledgements from those TRILL switches are returned to the end station by the proxy. Because native RBridge Channel messages have no TRILL Header and are addressed by MAC address, as opposed to inter-RBridge Channel messages that are TRILL Data packets and are addressed by nickname, nickname information must be added to the native RBridge Channel version of Pull Directory messages.

The native Pull Directory RBridge Channel messages use the same Channel protocol number as do the inter-RBridge Pull Directory RBridge Channel messages. The native messages SHOULD be sent with an Outer.VLAN tag that gives the priority of each message which is the priority of the original inter-RBridge request packet. The Outer.VLAN

ID used is the Designated VLAN on the link to the end station

[RFC6325]. Since there is no TRILL Header or inner Data Label for native RBridge Chanel messages, that information is added to the header.

The native RBridge Channel message Pull Directory message protocol dependent data part is the same as for inter-RBridge Channel messages except that the 8-byte header described in Section 3.1 is expanded to 14 or 18 bytes as follows:

```
                           1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 3 3
       0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |  Ver  | Type  | Flags | Count |      Err      |    SubErr     |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                        Sequence Number                        |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |   Nickname  (2 bytes)         |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+...+-+
      |   Data Label ... (4 or 8 bytes)                               |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+...+-+
      | Type Specific Payload - variable length
      +-+-+- ...
```

Fields not described below are as in Section 3.1.

Nickname: The nickname of the original TRILL switch that is communicating with the end station Pull Directory. Usually this is a remote TRILL switch but it could be the TRILL switch to which the end station is attached. The proxy copies this from the ingress nickname when mapping a Query or Acknowledge Message to native form. It also takes this from a native Response or Update Message to be used as the egress of the inter-RBridge form on the message unless it is a flooded Update in which case a distribution tree is used.

Data Label: The Data Label that normally appears right after the Inner.MacSA of the an RBridge Channel Pull Directory message appears here in the native RBridge Channel message version. This might appear in a native Query Message, to be reflected in a Response Message, or it might appear in a native Update to be reflected in an Acknowledge Message.

3.6 Pull Directory Message Errors

A non-zero Err field in the Pull Directory Reponse or Acknowledge Message header indicates an error message.

If there is an error that applies to an entire Query or Update

Message or its header, as indicated by the range of the value of the Err field, then the QUERY Records probably were not even looked at by the Pull Directory server and would provide no information in the Response or Acknowledge Message so they are omitted and the Count field is set to zero in the Response or Acknowledgement Message.

If errors occur at the QUERY Record level for a Query Message, they MUST be reported in a Response Message separate from the results of any successful non-erroneous QUERY Records. If multiple QUERY Records in a Query Message have different errors, they MUST be reported in separate Response Messages. If multiple QUERY Records in a Query Message have the same error, this error response MAY be reported in one or multiple Response Messages. In an error Response Message, the QUERY Record or Records being responded to appear, expanded by the Lifetime for which the server thinks the error might persist and with their Index inserted, as the RESPONSE Record or Records.

If errors occur at the RESPONSE Record level for an Update Message, they MUST be reported in a Acknowledge Message separate from the acknowledgement of any non-erroneous RESPONSE Records. If multiple RESPONSE Records in an Update have different errors, they MUST be reported in separate Acknowledge Messages. If multiple RESPONSE Records in an Update Message have the same error, this error response MAY be reported in one or multiple Acknowledge Messages. In an error Acknowledge Message, the RESPONSE Record or Records being responded to appear, expanded by the time for which the server thinks the error might persist and with their Index inserted, as a RESPONSE Record or Records.

ERR values 1 through 126 are available for encoding Request or Update Message level errors. ERR values 128 through 254 are available for encoding QUERY or RESPONSE Record level errors. The SubErr field is available for providing more detail on errors. The meaning of a SubErr field value depends on the value of the Err field.

3.6.1 Error Codes

Err	Level	Meaning
0	-	(no error)
1	Message	Unknown or reserved Query Message field value
2	Message	Request Message/data too short
3	Message	Unknown or reserved Update Message field value
4	Message	Update Message/data too short
5-126	Message	(Available for allocation by IETF Review)
127	-	Reserved
128	Record	Unknown or reserved QUERY Record field value
129	Record	QUERY Record truncated
130	Record	Address not found
131	Record	Unknown or reserved RESPONSE Record field value
132	Record	RESPONSE Record truncated
133-254	Record	(Available for allocation by IETF Review)
255	-	Reserved

Note that some error codes are for overall message level errors while some are for errors in the repeating records that occur in messages.

3.6.2 Sub-Errors Under Error Codes 1 and 3

The following sub-errors are specified under error code 1 and 3:

SubErr	Field with Error
0	Unspecified
1	Unknown Ver field value
2	Unknown Type field value
3	Specified Data Label not being served
4-254	(Available for allocation by Expert Review)
255	Reserved

3.6.3 Sub-Errors Under Error Codes 128 and 131

The following sub-errors are specified under error code 128 and 131:

SubErr	Field with Error
0	Unspecified
1	Unknown AFN field value
2	Unknown or Reserved QTYPE field value
3	Invalid or inconsistent SIZE field value
4	Invalid frame for QTYPE 2, 3, 4, or 5
5-254	(Available for allocation by Expert Review)
255	Reserved

3.7 Additional Pull Details

If a TRILL switch notices that a Pull Directory server is no longer data reachable [rfc7180bis], it MUST promptly discard all pull responses it is retaining from that server as it can no longer receive cache consistency Update Messages from the server.

A secondary Pull Directory server is one that obtains its data from a primary directory server. See discussion of primary to secondary directory information transfer in Section 2.5.

3.8 The No Data Flag

In the TRILL base protocol [RFC6325] as extended for FGL [RFC7172], the mere presence of an Interested VLANs or Interested Labels sub-TLVs in the LSP of a TRILL switch indicates connection to end stations in the VLAN(s) or FGL(s) listed and thus a desire to receive multi-destination traffic in those Data Labels. But, with Push and Pull Directories, advertising that you are a directory server requires using these sub-TLVs to indicate the Data Label(s) you are serving. If such a directory server does not wish to received multi-destination TRILL Data packets for the Data Labels it lists in one of these sub-TLVs, it sets the "No Data" (NOD) bit to one. This means that data on a distribution tree may be pruned so as not to reach the "No Data" TRILL switch as long as there are no TRILL switches interested in the Data that are beyond the "No Data" TRILL switch on the distribution tree. The NOD bit is backwards compatible as TRILL switches ignorant of it will simply not prune when they could, which is safe although it may cause increased link utilization.

Example of a TRILL switch serving as a directory that might not want multi-destination traffic in some Data Labels would be a TRILL switch that does not offer end station service for any of the Data Labels for which it is serving as a directory and is either

- a Pull Directory and/or

- a Push Directory for which all of the ESADI traffic will be handled by unicast ESDADI [RFC7357].

A Push Directory MUST NOT set the NOD bit for a Data Label if it needs to communicate via multi-destination ESADI PDUs in that data label since such PDUs look like TRILL Data packets to transit TRILL switches and are likely to be incorrectly pruned if NOD was set.

4. Directory Use Strategies and Push-Pull Hybrids

For some edge nodes that have a great number of Data Labels enabled, managing the MAC and Data Label <-> Edge RBridge mapping for hosts under all those Data Labels can be a challenge. This is especially true for Data Center gateway nodes, which need to communicate with many, if not all, Data Labels.

For those edge TRILL switch nodes, a hybrid model should be considered. That is, the Push Model is used for some Data Labels or addresses within a Data Label while the Pull Model is used for other Data Labels or addresses within a Data Label. It is the network operator's decision by configuration as to which Data Labels' mapping entries are pushed down from directories and which Data Labels' mapping entries are pulled.

For example, assume a data center where hosts in specific Data Labels, say VLANs 1 through 100, communicate regularly with external peers. Probably, the mapping entries for those 100 VLANs should be pushed down to the data center gateway routers. For hosts in other Data Labels that only communicate with external peers occasionally for management interfacing, the mapping entries for those VLANs should be pulled down from directory when the need comes up.

Similarly, it could be that within a Data Label that some addresses, such as the addresses of gateways, file, DNS, or database server hosts are commonly referenced by most other hosts but those other hosts, perhaps compute engines, are typically only referenced by a few hosts in that Data Label. In that case, the address information for the commonly referenced hosts could be pushed as an incomplete directory while the addresses of the others are pulled when needed.

The mechanisms described above for Push and Pull Directory services make it easy to use Push for some Data Labels or addresses and Pull for others. In fact, different TRILL switches can even be configured so that some use Push Directory services and some use Pull Directory services for the same Data Label if both Push and Pull Directory services are available for that Data Label. And there can be Data Labels for which directory services are not used at all.

There are a wide variety of strategies that a TRILL switch can adopt for making use of directory assistance. A few suggestions are given below.

- Even if a TRILL switch will normally be operating with information from a complete Push Directory server, there will be a period of time when it first comes up before the information it holds is complete. Or, it could be that the only Push Directories that can push information to it are incomplete or that they are just starting and may not yet have pushed the entire directory.

Thus, it is RECOMMENDED that all TRILL switches have a strategy for dealing with the situation where they do not have complete directory information. Examples are to send a Pull Directory query or to revert to [RFC6325] behavior.

- If a TRILL switch receives a native frame X resulting in seeking directory information, a choice needs to be made as to what to do if it does not already have the directory information it needs. In particular, it could (1) immediately flood the TRILL Data packet resulting from ingressing X in parallel with seeking the directory information, (2) flood that TRILL Data packet delayed, if it fails to obtain the directory information, or (3) discard X if it fails to obtain the information. The choice might depend on the priority of frame X since the higher that priority, the more urgent the frame is and the greater the probability of harm in delaying it. If a Pull Directory request is sent, it is RECOMMENDED that its priority be derived from the priority of the frame X with the derived priority configurable and having the following defaults:

Ingressed Priority	If Flooded Immediately	If Flooded After Delay
7	5	6
6	5	6
5	4	5
4	3	4
3	2	3
2	0	2
0	1	0
1	1	1

Priority 7 is normally only used for urgent messages critical to adjacency and so SHOULD NOT be the default for directory traffic. Unsolicited updates are sent with a priority that is configured per Data Label that defaults to priority 5.

5. Security Considerations

Incorrect directory information can result in a variety of security threats including the following:

> Incorrect directory mappings can result in data being delivered to the wrong end stations, or set of end stations in the case of multi-destination packets, violating security policy.
>
> Missing or incorrect directory data can result in denial of service due to sending data packets to black holes or discarding data on ingress due to incorrect information that their destinations are not reachable.

Push Directory data is distributed through ESADI-LSPs [RFC7357] that can be authenticated with the same mechanisms as IS-IS LSPs. See [RFC5304] [RFC5310] and the Security Considerations section of [RFC7357].

Pull Directory queries and responses are transmitted as RBridge-to-RBridge or native RBridge Channel messages [RFC7178]. Such messages can be secured as specified in [ChannelTunnel].

For general TRILL security considerations, see [RFC6325].

6. IANA Considerations

This section gives IANA assignment and registry considerations.

6.1 ESADI-Parameter Data Extensions

Action 1: IANA will assign a two bit field [bits 1-2 suggested] within the ESADI-Parameter TRILL APPsub-TLV flags for "Push Directory Server Status" (PDSS) and will create a sub-registry in the TRILL Parameters Registry as follows:

Sub-Registry: ESADI-Parameter APPsub-TLV Flag Bits

Registration Procedures: Standards Action

References: [RFC7357] [This document]

Bit	Mnemonic	Description	Reference
0	UN	Supports Unicast ESADI	ESDADI [RFC7357]
1-2	PDSS	Push Directory Server Status	[this document]
3-7	-	Available for assignment	

Action 2: In addition, the ESADI-Parameter APPsub-TLV is optionally extended, as provided in its original specification in ESADI [RFC7357], by one byte as show below. Therefore [this document] should be added as a second reference to the ESDAI-Parameter APPsub-TLV in the "TRILL APPsub-TLV Types under IS-IS TLV 251 Application Identifier 1" Registry.

```
+-+-+-+-+-+-+-+-+
| Type          |           (1 byte)
+-+-+-+-+-+-+-+-+
| Length        |           (1 byte)
+-+-+-+-+-+-+-+-+
|R| Priority    |           (1 byte)
+-+-+-+-+-+-+-+-+
| CSNP Time     |           (1 byte)
+-+-+-+-+-+-+-+-+
| Flags         |           (1 byte)
+---------------+
|PushDirPriority|           (optional, 1 byte)
+---------------+
| Reserved for expansion    (variable)
+-+-+-+-...
```

The meanings of all the fields are as specified in ESDADI [RFC7357] except that the added PushDirPriority is the priority of the advertising ESADI instance to be a Push Directory as described in Section 2.3. If the PushDirPriority field is not present (Length = 3) it is treated as if it were 0x40. 0x40 is also the value used and placed here by an TRILL switch whose priority to be a Push Directory has not been configured.

6.2 RBridge Channel Protocol Number

Action 3: IANA will allocate a new RBridge Channel protocol number for "Pull Directory Services" from the range allocable by Standards Action and update the subregistry of such protocol number in the TRILL Parameters Registry referencing this document.

6.3 The Pull Directory (PUL) and No Data (NOD) Bits

Action 4: IANA is requested to assign a currently reserved bit in the Interested VLANs field of the Interested VLANs sub-TLV [suggested bit 18] and the Interested Labels field of the Interested Labels sub-TLV [suggested bits 6] [RFC7176] to indicate Pull Directory server (PUL). This bit is to be added, with this document as reference, to the "Interested VLANs Flag Bits" and "Interested Labels Flag Bits" subregistries created by [RFC7357].

Action 5: IANA is requested to assign a currently reserved bit in the Interested VLANs field of the Interested VLANs sub-TLV [suggested bits 19] and the Interested Labels field of the Interested Labels sub-TLV [suggested bits 7] [RFC7176] to indicate No Data (NOD, see Section 3.8). This bit is to be added, with this document as reference, to the "Interested VLANs Flag Bits" and "Interested Labels Flag Bits" subregistries created by [RFC7357].

6.4 TRILL Pull Directory QTYPEs

Action 6: IANA is requested to create a new Registry on the "Transparent Interconnection of Lots of Links (TRILL) Parameters" web page as follows:

Name: TRILL Pull Directory QTYPEs"
Registration Procedure: IETF Review
Reference: [this document]
Initial contents as in Section 3.2.1.

6.5 Pull Directory Error Code Registries

Actions 7, 8, and 9: IANA is requested to create a new Registry and two new SubRegistries on the "Transparent Interconnection of Lots of Links (TRILL) Parameters" web page as follows:

```
Registry
   Name: TRILL Pull Directory Errors
   Registration Procedure: IETF Review
   Reference: [this document]

   Initial contents as in Section 3.6.1.

   Sub-Registry
      Name: Sub-codes for TRILL Pull Directory Errors 1 and 3
      Registration Procedure: Expert Review
      Reference: [this document]

      Initial contents as in Section 3.6.2.

   Sub-Registry
      Name: Sub-codes for TRILL Pull Directory Errors 128 and 131
      Registration Procedure: Expert Review
      Reference: [this document]

      Initial contents as in Section 3.6.3.
```

Normative References

[RFC826] - Plummer, D., "An Ethernet Address Resolution Protocol", RFC 826, November 1982.

[RFC903] - Finlayson, R., Mann, T., Mogul, J., and M. Theimer, "A Reverse Address Resolution Protocol", STD 38, RFC 903, June 1984

[RFC2119] - Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997

[RFC3971] - Arkko, J., Ed., Kempf, J., Zill, B., and P. Nikander, "SEcure Neighbor Discovery (SEND)", RFC 3971, March 2005.

[RFC4861] - Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, September 2007.

[RFC5304] Li, T. and R. Atkinson, "IS-IS Cryptographic Authentication", RFC 5304, October 2008.

[RFC5310] - Bhatia, M., Manral, V., Li, T., Atkinson, R., White, R., and M. Fanto, "IS-IS Generic Cryptographic Authentication", RFC 5310, February 2009.

[RFC6165] - Banerjee, A. and D. Ward, "Extensions to IS-IS for Layer-2 Systems", RFC 6165, April 2011.

[RFC6325] - Perlman, R., Eastlake 3rd, D., Dutt, D., Gai, S., and A. Ghanwani, "Routing Bridges (RBridges): Base Protocol Specification", RFC 6325, July 2011.

[RFC7042] - Eastlake 3rd, D. and J. Abley, "IANA Considerations and IETF Protocol and Documentation Usage for IEEE 802 Parameters", BCP 141, RFC 7042, October 2013.

[RFC7172] - Eastlake 3rd, D., Zhang, M., Agarwal, P., Perlman, R., and D. Dutt, "Transparent Interconnection of Lots of Links (TRILL): Fine-Grained Labeling", RFC 7172, May 2014, <http://www.rfc-editor.org/info/rfc7172>.

[RFC7176] - Eastlake 3rd, D., Senevirathne, T., Ghanwani, A., Dutt, D., and A. Banerjee, "Transparent Interconnection of Lots of Links (TRILL) Use of IS-IS", RFC 7176, May 2014, <http://www.rfc-editor.org/info/rfc7176>.

[RFC7178] - Eastlake 3rd, D., Manral, V., Li, Y., Aldrin, S., and D. Ward, "Transparent Interconnection of Lots of Links (TRILL):

RBridge Channel Support", RFC 7178, May 2014, <http://www.rfc-

editor.org/info/rfc7178>.

[RFC7357] - Zhai, H., Hu, F., Perlman, R., Eastlake 3rd, D., and O. Stokes, "Transparent Interconnection of Lots of Links (TRILL): End Station Address Distribution Information (ESADI) Protocol", RFC 7357, September 2014, <http://www.rfc-editor.org/info/rfc7357>.

[rfc7180bis] - D. Eastlake 3rd, M. Zhang, A. Banerjee, A. Ghanwani, and S. Gupta "Transparent Interconnection of Lots of Links (TRILL): Clarifications, Corrections, and Updates", RFC 7180, May 2014, <http://www.rfc-editor.org/info/rfc7180>.

[IA] - Eastlake, D., L. Yizhou, R. Perlman, "TRILL: Interface Addresses APPsub-TLV", draft-ietf-trill-ia-appsubtlv, work in progress.

Informational References

[RFC7067] - Dunbar, L., Eastlake 3rd, D., Perlman, R., and I. Gashinsky, "Directory Assistance Problem and High-Level Design Proposal", RFC 7067, November 2013.

[ChannelTunnel] - D. Eastlake, M. Umair, Y. Li, "TRILL: RBridge Channel Tunnel Protocol", draft-ietf-trill-channel-tunnel, work in progress.

[ARPreduction] - Y. Li, D. Eastlake, L. Dunbar, R. Perlman, I. Gashinsky, "TRILL: ARP/ND Optimization", draft-ietf-trill-arp-optimization, work in progress.

Acknowledgments

The contributions of the following persons are gratefully acknowledged:

Gsyle Noble

The document was prepared in raw nroff. All macros used were defined within the source file.

Authors' Addresses

Donald Eastlake
Huawei Technologies
155 Beaver Street
Milford, MA 01757 USA

Phone: +1-508-333-2270
Email: d3e3e3@gmail.com

Linda Dunbar
Huawei Technologies
5430 Legacy Drive, Suite #175
Plano, TX 75024, USA

Phone: +1-469-277-5840
Email: ldunbar@huawei.com

Radia Perlman
EMC
2010 256th Avenue NE, #200
Bellevue, WA 98007 USA

Email: Radia@alum.mit.edu

Igor Gashinsky
Yahoo
45 West 18th Street 6th floor
New York, NY 10011

Email: igor@yahoo-inc.com

Yizhou Li
Huawei Technologies
101 Software Avenue,
Nanjing 210012 China

Phone: +86-25-56622310
Email: liyizhou@huawei.com

Copyright, Disclaimer, and Additional IPR Provisions