INTERNET-DRAFT
Intended status: Proposed Standard
Donald Eastlake
Linda Dunbar
Huawei
Radia Perlman
EMC
Yizhou Li
Huawei
Expires: September 1, 2017
March 2, 2017

TRILL: Edge Directory Assist Mechanisms

<draft-ietf-trill-directory-assist-mechanisms-12.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.

Table of Contents

Table of Contents Continued

1. Introduction

[RFC7067] gives a problem statement and high level design for using directory servers to assist TRILL [RFC6325] [RFC7780] edge nodes in reducing multi-destination ARP/ND [ARPND], 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 TRILL directory servers.

The information held by the Directory(s) is address mapping and reachability information. Most commonly, what MAC (Media Access Control) 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 port on that TRILL switch, from which that MAC address is reachable. But it could be what IP address corresponds to a MAC address or possibly other address mapping or reachability information.

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 2.6. 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.

Efficient routing of unicast traffic in a TRILL campus assumes that the mapping of destination MAC addresses to edge RBridges is stable enough that the default data plane learning of TRILL and/or the use of directories reduces to an acceptable level the need to flood packets where the location of the destination is unknown. Although not prohibited, "Ephemeral" MAC addresses are unlikely to be used in such an environment. Directories need not be complete and in the case that any ephemeral MAC addresses were in use, they would probably not be included in directory information.

Directory services can be offered in a Push Mode, Pull Mode, or both [RFC7067] at the option of the server. 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.

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 switch could intercept ARP messages and reply if the TRILL switch has the relevant information [ARPND].

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, a directory assisted edge TRILL switch could intercept ND messages and reply if the TRILL switch has the relevant information. [ARPND]

3. Unknown destination MAC addresses normally cause a native frame to be flooded. 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 have learned that information from the directory or could query the directory if it does not know it. Furthermore, if the edge TRILL switch has complete directory information, it can detect a forged source MAC or IP address in any native frame and discard the frame if it finds such a forged address.

4. RARP [RFC903] (Reverse ARP) 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:

AFN: Address Family Number, (http://www.iana.org/assignments/address-family-numbers/)

CSNP Time: Complete Sequence Number PDU Time. See ESDADI [RFC7357] and Section 7.1 below.

Data Label: VLAN or FGL.

ESADI: End Station Address Distribution Information [RFC7357].

FGL: Fine Grained Label [RFC7172].

FR: Flood Record flag bit. See Section 3.2.1.

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

Interested Labels sub-TLV: Short for "Interested Labels and Spanning Tree Roots sub-TLV" [RFC7176].

Interested VLANs sub-TLV: Short for "Interested VLANs and Spanning Tree Roots sub-TLV" [RFC7176].

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

MAC: Media Access Control address [RFC7042]

MacDA: Destination MAC address.

MacSA: Source MAC address.

OV: Overflow flag bit. See Section 3.2.2.1.

PDSS: Push Directory Server Status. See Sections 2 and 7.1.

PUL: Pull Directory flag bit. See Sections 3 and 7.3.

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.

TLV: Type, Length, Value

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 [RFC7961]. This service is scoped by Data Label (VLAN or FGL [RFC7172]). A Push Directory advertises when, for a Data Label, it both is configured to be a directory having complete information and has actually pushed all the information it has. 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] (End Station Address Distribution Information) 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) that 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 the Directory information to be pushed to them 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, through ESADI, their availability to push the mapping information for a particular Data Label by setting the PDSS (Push Directory Server Status) in their ESADI Parameter APPsub-TLV for that ESADI instance (see [RFC7357] and Section 7.1) to a non-zero value. This PDSS field setting is visible to other ESADI participants, including other Push Directory servers, for that Data Label. 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 increased robustness, increased bandwidth capability, and improved locality, 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 (see PushDirServers, Section 2.7). 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 be pushed.

Each Push Directory server also has a configurable 8-bit priority (PushDirPriority) to be Active, which defaults to 0x3F (see Section 2.7). This priority is treated as an unsigned integer where larger magnitude means higher priority. This priority appears in its ESADI Parameter APPsub-TLV (see Section 7.1). In case of a tie in this configurable priority, the System ID of the TRILL switch acting as the server is used as an unsigned 6-byte integer where larger magnitude indicates higher priority.

For each Data Label it can serve, each Push Directory server checks to see if there appear to be enough higher priority servers to push the desired number of copies. It does this by ordering, by priority, the Push Directory servers whose advertisements are present in the ESADI link state database for that Data Label and that are data reachable [RFC7780] as indicated by its IS-IS link state database. The Push Directory server then determines its own position in that order. If a Push Directory server's configuration indicates that N copies of the mappings for a Data Label should be pushed and the server finds that it is number K in the priority ordering (where number 1 in the ordered list is highest priority and the last is lowest priority), 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 in Section 2.3.

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 in ESADI 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 angle 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 that 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 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 2. It also advertises its directory data and any changes through ESADI [RFC7357] in its ESADI-LSPs using the Interface Addresses [RFC7961] APPsub-TLV and updates that information as it changes.

Active Completing <S4>: Same behavior as the Active state except that the server 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 (see the Time Condition, Section 2.3.2) 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 Was Complete <S6>: The same behavior as Active except that the server responds differently to events. The purpose of this state is to be sure that the information, that the directory will no longer be complete, has enough time to propagate to edge TRILL switches (see the Time Condition, Section 2.3.2) before the Directory Server stops advertising updates to the information. (See note below.)

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 will no longer be complete, has enough time to propagate to edge TRILL switches (see the Time Condition, Section 2.3.2) 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 that was complete being configured to be incomplete and then the Stand-By Condition (see Section 2.3.2) occurring shortly 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 for Data Label X, this Push Directory server should be active. This is determined by the server finding that (A) it is priority K among the data reachable Push Directory servers (where highest priority server is 1) for Data Label X, (B) it is configured that there should be N copies pushed for Data Label X, and (C) K is less than or equal to N. For example, if the Push Directory server is configured so that 2 copies should be pushed and finds that it is priority 1 or 2 among the Push Directory servers that are visible in its ESADI link state database and that are data reachable as indicated by its IS-IS link state database.

The Stand-By Condition: In order to have the desired number of Push Directory servers pushing data for Data Label X, this Push Directory server should be stand-by (not active). This is determined by the server finding that (A) it is priority K among the data reachable Push Directory servers (where highest priority server is 1) for Data Label X, (B) it is configured that there should be N copies pushed for Data Label X, and (C) 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 available Push directory servers.

The Time Condition: The Push Directory server has been in its current state for a configurable amount of time (PushDirTimer) that defaults to twice its CSNP (Complete Sequence Number PDU) time (see Sections 2.7 and 7.1).)

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

1. Push Directory server comes Up.

2. The Push Directory server or the TRILL switch on which it resides is being shut down. This is a persistent condition unless the shut down is canceled. So, for example, a Push Directory server in the Going Stand-By Was Complete state does not transition out of that state due to this condition but, after the Time Condition is met and the directory transitions to Stand-By and performs the actions required there (such as purging LSPs) continues to the Down state if this condition is still true. Similar comments apply to

events/conditions 3, 4, and 5.

3. The Activate Condition is met and the server's configuration indicates it does not have complete data.

4. The Stand-By Condition is met.

5. The Activate Condition is met and the server's configuration indicates it has complete data.

6. The server's configuration is changed to indicate 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 / Event	Down <S1>	Stand-By <S2>	Active <S3>	Active Completing <S4>	Active Complete <S5>	Going Stand-By <S6>	Active Uncompleting <S7>
1	<S2>	N/A	N/A	N/A	N/A	N/A	N/A
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 Stand-By |
   |   Complete  |       |  Uncompleting  |    |  Was Complete  |
   |     <S5>    |<-------|      <S7>      |    |      <S6>      |
   +-------------+      5 +----------------+    +----------------+
    |1,5,7  ^  |2,4         |1,2,3,6     ^        ^   |1,2,4,6 ^
    |       |  |            |            |        |   |        |
    +-------+  |            +------------+        |   +--------+
               |                                  |
               +----------------------------------+
```

Figure 1. Push Server State Diagram

2.4 End Stations and Push Directories

End station hosting or use of Push Directories is outside of the scope of this document. Push Directory information distribution is accomplished using ESADI [RFC7357], which does not operate to end stations. In the future, ESADI might be extended to operate to end stations or some other method, such as BGP, might be specified as a way to support end station hosting or use of Push Directories.

2.5 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 [RFC7961] and are flagged in that APPsub-TLV as being Push Directory data.

TRILL switches, whether or not they are Push Directory servers, 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 TRILL Data packets 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 [RFC7780] as indicated by the IS-IS link state database, other 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 (PushDirTimer, see Section 2.7).

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 fragments from the Push Directory server.

Likewise, due to various delays, when an end station connects to or disconnects from the campus there are timing differences between such connection or disconnection, the update of directory information at the directory, and the update of directory information at any particular RBridge in the TRILL campus. Thus, there is commonly a small window during which an RBridge using directory information might either (1) drop or unnecessarily flood a frame as having an unknown unicast destination or (2) encapsulate a frame to an edge RBridge where the end station is not longer connected when the frame arrives at that edge RBridge.

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] [RFC7961] 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.

2.6 Primary to Secondary Server Push Service

A secondary Push or Pull Directory server is one that obtains its data from a primary directory server. Such mechanisms, where some directory servers can be populated from others, have been found useful for multiple-server directory applications, for example in the DNS where it is the normal case that some authoritative servers (secondary servers) are populated with data from other authoritative servers (primary servers).

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 at least one less than that of the data as received from the primary (or unchanged if it is already of minimum confidence).

2.7 Push Directory Configuration

The following per Data Label configuration parameters are available for controlling Push Directory behavior:

Name	Range	Default	Section
PushDirService	T/F	F	2.2
PushDirServers	1 - 8	2	2.2
PushDirPriority	0 - 255	0x3F	2.2
PushDirComplete	T/F	F	2.3.1, 2.3.2
PushDirTimer	1 - 511	2*CSNP	2.3.2, 2.5

PushDirService is a boolean. When false, Push Directory service is not provided; when true, it is.

PushDirComplete is a boolean. When false, the server never indicates that the information it has pushed is complete; when true, it does so indicate after pushing all the information it knows.

PushDirTimer defaults to two times the ESADI CSNP configuration value but not less than 1 second.

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.

A TRILL switch that makes use of Pull Directory services must implement appropriate connections between its directory utilization and its link state database and link state updating. For example, 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 [RFC7780] TRILL switches that advertise themselves by having the Pull Directory flag (PUL) on in their Interested VLANs or Interested Labels sub-TLV (see Section 7.3) for that Data Label. The set of such switches can change with configuration changes by network management, such as starting up or shutting down of Pull Directory servers, or changes in network topology, such the connection or disconnection of TRILL switches that are Pull Directory servers, or network partition or merger. As described in Section 3.7, a TRILL switch MUST notice if a Pull Directory from which it has cached data is no longer data reachable so it can discard such cached data.

If multiple data reachable TRILL switches indicate in the link state database 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 7.2). 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 are returned as described in Section 3.6.

The requests to Pull Directory Servers are typically derived from ingressed ARP [RFC826], ND [RFC4861], RARP [RFC903], or SEND [RFC3971] 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 be cached and used to locally handle ARP, ND, RARP, unknown destination MAC frames, or the like [ARPND], 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 unless it has 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 Channel Protocol specific 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. See Section 3.1.1 for a discussion of version negotiation.

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: Some 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 acknowledgment. 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.

3.1.1 Version Negotiation

The version number (Ver) in the Pull Directory message header is incremented for a future version with changes such that TRILL directory messages cannot be parsed correctly by an earlier version. Ver is not incremented for minor changes such as defining a new field value for an existing field.

Pull Directory messages come in pairs (Request-Response, Update-Acknowledgment). The version number in the Request/Update (Ver1) indicates the format of that message and of the corresponding returned Response/Acknowledgment. The version number in the returned Response/Acknowledgment (Ver2) indicates the highest version number that the sender of that Response/Acknowledgment understands.

In the most common case of a well configured network, Ver1 and Ver2 will be equal.

If Ver2 is less than Ver1, the returned Response/Acknowledgment will be an error message saying that the version is not understood.

If Ver2 is greater than Ver1 and the responder understands Ver1, it responds normally in Ver1 format. However, if the responder does not understand Ver1, it MUST send a version-not-understood error message correctly formatted for Ver1. Thus all implementations that support some version X MUST be able to send a version-not-understood error message formatted correctly formatted for all lower versions down to version zero.

3.2 Pull Directory Query and Response Messages

The format of 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 ingressed frame that caused the query. The default mapping depends, per Data Label, on the strategy (see Section 4) or a configured priority (DirGenQPriority, Section 3.9) 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 Section 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 implemented or 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 returned 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          |FR|  RESV  |   QTYPE   |
    +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
 If QTYPE = 1
    +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
    |                      AFN                      |
    +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
    |  Query address ...
    +--+--+--+--+--+--+--+--+--+--+--+--...
 If QTYPE = 2 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.

FR: The Flood Record flag that is ignored if QTYPE is zero. If QTYPE is 2 or 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. For QTYPEs other than 2 or 5, the FR flag has no effect.

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	Frame query
3-4	Unassigned
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 as this address, within the Data Label of the query of the address provided. A typically Query Address would be something like the following:

(1) A 48-bit MAC address with the querying TRILL switch primarily interested in either
   - (1a) the RBridge by which that MAC address is reachable so that the querying RBridge can forward an unknown (before the query) destination MAC address native frame as a unicast TRILL Data packet rather than flooding it, or
   - (1b) the IP address corresponding to the MAC address so that RBridge can locally respond to a RARP [RFC903] native frame.

(2) An IPv4 or IPv6 address with the querying RBridge interested in the corresponding MAC address so it can locally respond to an ARP [RFC826] or ND [RFC4861] native frame [ARPND].

But the query address could be some other address type for which an AFN has been assigned, such as a 64-bit MAC address [RFC7042] or a CLNS address [X.233].

Query Frame: Where a QUERY Record is the result of an ARP, ND, RARP, SEND, 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. The full frame is included, starting with the destination and source MAC addresses but does not include the FCS.

If no response is received to a Pull Directory Query Message within a configurable timeout (DirQueryTimeout, see Section 3.9), then the

Query Message should be re-transmitted with the same Sequence Number (up to a configurable number of times (DirQueryRetries, see Section

3.9)). 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 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 the configured value DirRespMaxPriority (Section 3.9).

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 RESPONSE Record. The RESPONSE Record with such an excessive SIZE value and any subsequent RESPONSE Records MUST be ignored and the entire Response 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 Response Message has a message level error code in it, then the RESPONSE Records are omitted and Count will be zero. See Section 3.6 for additional information on errors.
- If the Err field of the enclosing 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 Response Message is zero, then the Response Data in each RESPONSE Record is formatted as the value part of an Interface Addresses APPsub-TLV [RFC7961]. The maximum size of such contents is 255 bytes, in which case the RESPONSE Record SIZE field is 255.

Multiple RESPONSE Records can appear in a Response Message with the same Index if an answer to the 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 and 5 are interpreted and handled as described below. In these cases, if the information implicitly sought is not in the directory and the FR flag in the query message was one, the provided frame is forwarded by the Pull Directory server as a multi-destination TRILL Data packet with the ingress nickname of the Pull Directory server (or proxy if it is hosted on an end station) in the TRILL header. If the FR flag is zero, the frame is not forwarded in this case.

If there was no error in the handling of the enclosing Query Message, 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 and the Ethertype in the QUERY Record indicates that an ARP [RFC826] frame is included in the 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/SEND: When QTYPE is 2 and the Ethertype in the QUERY Record indicates an IPv6 Neighbor Discover (ND [RFC4861]) or Secure Neighbor Discover (SEND [RFC3971]) frame is included in the 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, if the ND is not a SEND, an ND Neighbor Advertisement response is returned by the Pull Directory server as a TRILL Data packet. In the case of SEND [RFC3971], an error is returned indicating that SEND was received by the Pull Directory and the Pull Directory then either forwards the SEND frame to the holder of the IPv6 address if that information is in the directory or the directory multicasts the SEND frame.

RARP: When QTYPE is 2 and the Ethertype in the QUERY Record indicates that a RARP [RFC903] frame is included in the 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 frame is provided in the QUERY Record whose destination MAC address TRILL switch attachment is unknown, the only requirement is that this MAC address has to be unicast. The Ethertype in the QUERY Record is ignored. 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 three sets of records concerning possible cached data at clients of that server. These are numbered and listed below in order of increasing specificity:

Method 1, Least Specific. An overall record per Data Label of when the last positive response data sent will expire and when the last negative response sent will expire; the records are retained until such expiration.
Pro: Minimizes the record keeping burden on the Pull Directory server.
Con: Increases the volume of and overhead due to spontaneous Update Messages and due to unnecessarily invalidating cached information.

Method 2, Medium Specificity. For each unit of data (IA APPsub-TLV Address Set [RFC7961]) held by the server, record when the last response sent with that positive response data will expire. In addition, record each address about which a negative response was sent by the server and when the last such negative response will expire. Each such record of a positive or negative response is discarded upon expiration.
Pro/Con: An intermediate level of detail in server record keeping and an intermediate volume of and overhead due to spontaneous Update Messages with some unnecessary invalidation of cached information.

Method 3, Most Specific. For each unit of data held by the server (IA APPsub-TLV Address Set [RFC7961]) 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. Each list entry is retained until such expiration time.

Pro: Minimizes spontaneous Update Messages sent to update pull client TRILL switch caches and minimizes unnecessary invalidation of cached information.

Con: Increased record keeping burden on the Pull Directory server.

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

A Pull Directory server might have a limit as to how many TRILL switches for which it can maintain detailed 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, and SHOULD support methods 2 and 3. Use of method 1 may be quite inefficient due to large amounts of cached positive and negative information being unnecessarily discarded.

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 (DirUpdateDelay, see Section 3.9) to await other possible changes that could be included in the same Update.

1. If method 1, the least detailed 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. Similarly if data is added and there are outstanding cached negative responses, an all-addresses flush negative message is similarly flooded. The Count field is 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 (flood), P (positive), and N (negative) bits (see Section 3.3.1) set to one and the Count field zero 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 sent as RBridge Channel messages. The F bit will be one; 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 that might still be cached is updated:
   An unsolicited Update Message is sent with the P flag set and the Err field zero. On receipt, the addresses in the RESPONSE Records are compared to the addresses for which the receiving TRILL switch is holding cached positive information from that server. If they match, the cached information is updated.

   2.b If data that might still be cached is deleted:
   An unsolicited 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 do include an interface address set. Any cached positive information for the addresses given is deleted and the negative response is cached as per the lifetime given.

   2.c If data for an address for which a negative response was sent is added, so that negative response that might still be cached is now incorrect:
   An unsolicited Update Message is sent with the N flag set to one and the Err field zero. The addresses in the RESPONSE Records are compared to the addresses for which the receiving TRILL switch is holding cached negative information from that server; 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 negative information that might

still be cached:
An unsolicited Update Message is sent 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 deletion or 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 so that flooding is more efficient that unicast.

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 either (a) sent an Update Message and received a corresponding Acknowledge Message or (b) it has sent a configurable number of updates at a configurable interval that default to 3 updates 100 milliseconds apart (see Section 3.9).

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 number of updates at an interval as in the paragraph above.

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 Index field in the RESPONSE Record(s) is set to zero on transmission and ignored on receipt (but the Count field in the Update Message header MUST still correctly indicate the number of RESPONSE Records present).
- The priority with which the message is sent, DirUpdatePriority, is configurable and defaults to 5 (see Section 3.9).

Update Messages are initiated by a Pull Directory server. The Sequence number space used is controlled by the originating Pull

Directory server. This update Sequence number space is different from the 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. If both are set, the Update Message MUST be ignored as this combination is only valid for method 1.

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 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 echoed back with the Err and SubErr fields set appropriately, 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 as discussed in Section 3.6.

The Acknowledge Messages is sent with the same priority as the Update Message it acknowledges but not more than a configured priority (DirAckMaxPriority) that defaults to 5 (see Section 3.9).

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 abbreviations in this table have the meanings listed below. "IA APPsub-TLV value" means the value part of the IA APPsub-TLV specified in [RFC7961].

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 APPsub-TLV 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 APPsub-TLV 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 End Stations and Pull Directories

A Pull Directory can be hosted on an end station as specified in Section 3.5.1.

A end station can use a Pull Directory as specified in Section 3.5.2. This capability would be useful in supporting an end station that performs directory assisted encapsulation [DirAsstEncap] or that is a "smart end node" [SmartEN].

The native Pull Directory messages used in these cases are as specified in Section 3.5.3. In these cases, the edge RBridge(s) and end station(s) involved need to detect each other and exchange some control information. This is accomplished with the TRILL ES-IS mechanism specified in Section 5.

3.5.1 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 act as indirect Pull Directory servers. That is, they host a Pull Directory server, which is seen by other TRILL switches in the campus, and a Pull Directory client, which fetches directory information from one or more End Station Pull Directory servers, where at least some of the information served up by the Pull Directory server may be information fetched from an end station to which it is directly connected by the co-located Pull Directory client. (Direct connection means a connection not involving any intermediate TRILL switches.)

End stations hosting a Pull Directory server MUST support TRILL ES-IS (see Section 5) and advertise the Data Labels for which they are providing service in one or more Interested VLAN or Interested Label sub-TLVs by setting the PUL flag (see Section 7.3).

```
                                                *  *  *  *  *  *  *
      +---------------+                         *                 *
      | End Station 1 |              +---------------+            *
      | Pull Directory+--------------+ RB1, Pull     |            *
      | Server        |              |      Directory|            *
      +---------------+      +-------+ Client|Server |         +-----+
                             |       +---------------+         |RB99|
      +---------------+      |                  *              +-----+
      | End Station 2 |   +--+---+   +---------------+            *
      | Pull Directory+---+Bridge+---+ RB2, Pull     |            *
      | Server        |   +--+---+   |      Directory|            *
      +---------------+      |       | Client|Server |            *
                             |       +---------------+            *
                             |                  *        TRILL    *
                             .                  *        Campus   *
                             .                  *                 *
                             .                  *  *  *  *  *  *  *
```

Figure 2. End Station Pull Directory Example

The figure above gives an example where RB1 and RB2 advertise themselves to the rest of the TRILL campus, such as RB99, as Pull Directory servers and obtain at least some of the information they are serving up by issuing Pull Directory queries to end stations 1 and/or 2. This example is specific but many variations are possible. The Bridge shown might be a complex bridged LAN, a LAN without a bridge (as shown for End Station 1), or connected via point-to-point links (as shown for End Station 2 that is connected through a bridge with point-to-point Ethernet links to RB1 and RB2). There could be one or more than two RBridges having such indirect Pull Directory servers. Furthermore, there could be one or more than two end stations with Pull Directory servers on them. Each TRILL switch server could then be differently configured as to the Data Labels for which it is providing indirect service selected from the union of the Data Labels supported by the End Station hosted servers and could select from among those End Station hosted servers supporting each Data Label the indirect server is configured to serve up.

When an indirect Pull Directory server receives a Query Message from another TRILL switch, it answers from information it has cached or issues Pull Directory request to End Station Pull Directory servers with which it has TRILL ES-IS adjacency to obtain the information. Any Response sent by an indirect Pull Directory server MUST NOT have a validity time longer that the valid of the data on which it is based. When an indirect Pull Directory server receives Update Messages, it updates its cached information and MUST originate Update messages to any clients that may have mirrors of the cached information so updated.

Since an indirect Pull Directory server discards information it has cached from queries to an end station Pull Directory server if it loses adjacency to the server (Section 3.7), if it detects that such information may be cached at RBridge clients and has no other source for the information, it MUST send Update Messages to those clients withdrawing the information. For this reason, indirect Pull Directory servers may wish to query multiple sources, if available, and cache multiple copies of returned information from those multiple sources. Then if one end station source becomes inaccessible or withdraws the information but the indirect Pull Directory server has the information from another source, it need not originate Updates.

3.5.2 Use of Pull Directory by End Stations

Some special end stations, such as those discussed in [DirAsstEncap] and [SmartEN], may need to access directory information. How edge RBridges provide this optional service is specified below.

When Pull Directory support is provided by an edge RBridge to end stations, the messages used are as specified in Section 3.5.3 below. The edge RBridge MUST support TRILL ES-IS (Section 5) and advertises the Data Labels for which it offers this service to end stations by setting the Pull Directory flag (PUL) to one in its Interested VLANs or Interested Labels sub-TLV (see Section 7.3) for that Data Label advertised through TRILL ES-IS.

3.5.3 Native Pull Directory Messages

The Pull Directory messages used between TRILL switches and end stations are native RBridge Channel messages [RFC7178]. These RBridge Channel messages use the same Channel protocol number as the inter-RBridge Pull Directory RBridge Channel messages. The Outer.VLAN ID used is the TRILL ES-IS Designated VLAN (see Section 5) on the link to the end station. Since there is no TRILL Header or inner Data Label for native RBridge Channel messages, that information is added to the Pull Directory message header as specified below.

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 12 or 16 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                        |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |   Data Label ... (4 or 8 bytes)                               |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+...+-+
  | Type Specific Payload - variable length
  +-+-+-+- ...
```

Fields other than Data Label are as in Section 3.1. The Data Label that normally appears right after the Inner.MacSA of the an RBridge Channel Pull Directory message appears in the Data Label field of the Pull Directory message header in the native RBridge Channel message version. This Data Label appears 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. Since the appropriate VLAN or FGL [RFC7172] Ethertype is included, the length of the Data Label can be determined from the first two bytes.

3.6 Pull Directory Message Errors

A non-zero Err field in the Pull Directory Response 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 additional information in the Response or Acknowledge Message. Therefore, the Records section of the Query Response or Update Message is omitted and the Count field is set to zero in the Response or Acknowledgment 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 (usually 2**16-1 which indicates indefinitely) 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 an Acknowledge Message separate from the acknowledgment 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

The following table lists error code values for the Err field, their meaning, and whether they apply at the Message or Record level.

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	Unassigned
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	Unassigned
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 codes 1 and 3:

SubErr	Field with Error
0	Unspecified
1	Version not understood (see Section 3.1.1)
2	Unknown Type field value
3	Specified Data Label not being served
4-254	Unassigned
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 (other than SEND)
5	SEND frame sent as QTYPE 2
6	Invalid frame for QTYPE 5 (such as multicast MacDA)
7-254	Unassigned
255	Reserved

3.7 Additional Pull Details

A Pull Directory client MUST notice, by tracking link state changes, when a Pull Directory server is no longer accessible (data reachable [RFC7780] for the inter-RBridge case or TRILL ES-IS (Section 5) adjacent for end station to RBridge case), and 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 2.6.

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 need to receive multi-destination traffic in those Data Labels. However, with Pull Directories, advertising that you are a directory server requires using these sub-TLVs to indicate the Data Label(s) you are serving.

If a directory server does not wish to received multi-destination TRILL Data packets for the Data Labels it lists in one of the Interested VLAN or Interested FGL [RFC7172] sub-TLVs, it sets the "No Data" (NOD) bit to one (see Section 7.3). 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 Label that are beyond the "No Data" TRILL switch on that 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 by some sending multi-destination traffic where it is not needed.

Push Directories advertise themselves inside ESADI which normally requires the ability to send and receive multi-destination TRILL Data packets but can be implemented with serial unicast.

Examples 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 one or more Data Labels where all of the ESADI traffic for those Data Labels will be handled by unicast ESADI [RFC7357].

A Push Directory MUST NOT set the NOD bit for a Data Label if it needs to communicate via multi-destination ESADI or RBridge Channel 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.

3.9 Pull Directory Service Configuration

The following per RBridge scalar configuration parameters are available for controlling Pull Directory service behavior. In addition, there is a configurable per Data Label mapping from the priority of a native frame being ingress to the priority of any Pull Directory query it causes. The default such mapping depends on the client strategy as described in Section 4.

Name	Default	Section	Note Below
DirQueryTimeout	100 milliseconds	3.2.1	1
DirQueryRetries	3	3.2.1	1
DirGenQPriority	5	3.2.1	2
DirRespMaxPriority	6	3.2.2.1	3
DirUpdateDelay	50 milliseconds	3.3	
DirUpdatePriority	5	3.3.1	
DirUpdateTimeout	100 milliseconds	3.3.3	
DirUpdateRetries	3	3.3.3	
DirAckMaxPriority	5	3.3.2	4

Note 1: Pull Directory Query client timeout waiting for response and maximum number of retries

Note 2: Priority for client generated requests (such as a query to refresh cached information).

Note 3: Pull Directory Response Messages SHOULD NOT be sent with priority 7 as that priority SHOULD be reserved for messages critical to network connectivity.

Note 4: Pull Directory Acknowledge Messages SHOULD NOT be sent with priority 7 as that priority SHOULD be reserved for messages critical to network connectivity.

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 in this document 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 after a delay, 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 typically 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

NOTE: The odd looking numbers towards the bottom of the columns above are because priority 1 is lower than priority zero. That is to say, the values in the first column are in priority order. They will look more logical if you think of "0" as being "1 1/2".

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. TRILL ES-IS

TRILL ES-IS (End System to Intermediate System) is a variation of TRILL IS-IS [RFC7176] [RFC7177] [RFC7780] designed to operate on a TRILL link among and between one or more TRILL switches and end stations on that link. Support of TRILL ES-IS is generally optional for both the TRILL switches and the end stations on a link but may be required to support certain features. As of the date of this document, the only features requiring TRILL ES-IS are those listed in this Section 5.

TRILL ES-IS is useful in supporting Pull Directory hosting on or use from end stations (see Section 3.5) and supporting specialized end stations [DirAsstEncap] [SmartEN] and may have additional future uses. The advantages of TRILL ES-IS over simply making an "end station" be a TRILL Switch include relieving the end station of having to maintain a copy of the core link state database (LSPs) and of having to perform routing calculations or having the ability to forward traffic.

Except as provided below in this Section 5, TRILL ES-IS PDUs and TLVs are the same TRILL IS-IS PDUs and TLVs.

5.1 PDUs and System IDs

All TRILL ES-IS PDUs (except some MTU-probe and MTU-ack PDUs which may be unicast) are multicast using the TRILL-ES-IS multicast MAC address (see Section 7.6). This use of a different multicast address assures that TRILL ES-IS and TRILL IS-IS PDUs will not be confused for one another.

Because end stations do not have IS-IS System IDs, TRILL ES-IS uses port MAC addresses in their place. This is convenient since MAC addresses are 48-bit and almost all IS-IS implementations use 48-bit System IDs. Logically TRILL IS-IS operates between the TRILL switches in a TRILL campus as identified by System ID while TRILL ES-IS operates between Ethernet ports on an Ethernet link (which may be a bridged LAN) as identified by MAC address [RFC6325].

As System IDs of TRILL Switches in a campus are required to be unique, so the MAC addresses of TRILL ES-IS ports on a link MUST be unique.

5.2 Adjacency, DRB Election, Hellos, TLVs, Etc.

TRILL ES-IS Adjacency formation and DRB election operate between the ports on the link as specified in [RFC7177] for a broadcast link. The DRB specifies an ES-IS Designated VLAN for the link. This adjacency determination, DRB election, and Designated VLAM are distinct from TRILL IS-IS adjacency, DRB election, and Designated VLAN.

Although the "Report State" [RFC7177] exists for TRILL ES-IS adjacencies, such adjacencies are only reported in TRILL ES-IS LSPs, not in any TRILL IS-IS LSPs.

End stations supporting TRILL ES-IS MUST assign a unique Port ID to each of their TRILL ES-IS ports which appears in the TRILL ES-IS Hellos they send.

TRILL ES-IS has nothing to do with Appointed Forwarders and the Appointed Forwarders sub-TLV and VLANs Appointed sub-TLV [RFC7176] are not used and SHOULD NOT be sent in TRILL ES-IS; if such a sub-TLV is received in TRILL ES-IS it is ignored. (The Appointed Forwarders on a link are determined as specified in [rfc6439bis] using TRILL IS-IS.)

Although some of the ports sending TRILL ES-IS PDUs are on end stations and thus not on routers (TRILL switches), they nevertheless may make use of the Router Capability (#242) and MT-Capability (#222) IS-IS TLVs to indicate capabilities as specified in [RFC7176].

TRILL ES-IS Hellos are like TRILL IS-IS Hellos but note the following: In the Special VLANs and Flags Sub-TLV, any TRILL switches advertise a nickname they own but for end stations that field MUST be sent as zero and ignored on receipt. In addition, the AF and TR flag bits MUST be sent as zero and the AC flag bit MUST be sent as one and all three are ignored on receipt.

5.3 Link State

The only link state transmission and synchronization that occurs in TRILL ES-IS is for E-L1CS PDUs (Extended Level 1 Circuit Scoped [RFC7356]). In particular, the end station Ethernet ports supporting TRILL ES-IS do not support the core TRILL IS-IS LSPs and do not support E-L1FS LSPs (or the CSNPs or PSNPs corresponding to either of them). TLVs and sub-TLVs that would otherwise be sent in TRILL IS-IS LSPs or E-L1FS SPs are instead sent in E-L1CS LSPs.

6. Security Considerations

For general TRILL security considerations, see [RFC6325].

6.1 Directory Information Security

Incorrect directory information can result in a variety of security threats including those below. Directory servers therefore need to take care to implement and enforce access control policies that are not overly permissive.

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, incorrect, or inaccessible 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 or that their source addresses are forged.

For these reasons directory information needs to be protected from unauthorized modification whatever server or end station it resides on. Parties authorized to modify directory data can violate availability and integrity policies.

6.2 Directory Confidentiality and Privacy

In implementations of the base TRILL protocol [RFC6325] [RFC7780], RBridges deal almost exclusively with MAC addresses. Use of directories to map to/from IP addresses means that RBridges deal more actively with IP addresses as well. But RBridges in any case would be exposed to plain text ARP/ND/SEND/IP traffic and so can see all this addressing meta-data. So this more explicit dealing with IP addresses has little effect on the privacy of end station traffic.

Parties authorized to read directory data can violate privacy polices for such data.

6.3 Directory Message Security Considerations

Push Directory data is distributed through ESADI-LSPs [RFC7357]. ESADI is built on IS-IS and such data can thus be authenticated with the widely implemented and deployed IS-IS PDU security. This mechanism provides authentication and integrity protection. 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 by the mechanisms specified in [RFC7978]. These mechanisms can provide authentication and confidentiality protections. At the time of this RFC, these security mechanisms are believed to be less widely implemented than IS-IS security.

7. IANA Considerations

This section gives IANA assignment and registry considerations.

7.1 ESADI-Parameter Data Extensions

Action 1: IANA is requested to 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	ESADI [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 ESADI-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              (variable)
|  expansion
+-+-+-+-...
```

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 0x3F. 0x3F is also the value used and placed here by an TRILL switch whose priority to be a Push Directory has not been configured.

7.2 RBridge Channel Protocol Numbers

Action 3: IANA is requested to assign a new RBridge Channel protocol number from the range assignable by Standards Action and update the subregistry of such protocol number in the TRILL Parameters Registry. Description is "Pull Directory Services". Reference is [this document].

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

Actions 4 and 5: IANA is requested to assign a currently reserved bits in the Interested VLANs field of the Interested VLANs sub-TLV and the Interested Labels field of the Interested Labels sub-TLV [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] as shown below after Action 7.

Actions 6 and 7: IANA is requested to assign a currently reserved bit in the Interested VLANs field of the Interested VLANs sub-TLV and the Interested Labels field of the Interested Labels sub-TLV [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] as shown below.

Bits and format suggested for above actions 4 through 7 are shown below:

Registry: Interested BLANs Flag Bits

Bit	Mnemonic	Description	Reference
18	PUL	Pull Directory	[this document]
19	NOD	No Data	[this document]

Registry: Interested Labels Flag Bits

Bit	Mnemonic	Description	Reference
6	PUL	Pull Directory	[this document]
7	NOD	No Data	[this document]

7.4 TRILL Pull Directory QTYPEs

Action 8: 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 Query Types (QTYPEs)
Registration Procedure: IETF Review
Reference: [this document]
Initial contents as in Section 3.2.1.

7.5 Pull Directory Error Code Registries

Actions 9, 10, and 11: IANA is requested to create a new Registry and two new indented SubRegistries under that Registry 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.

7.6 TRILL-ES-IS MAC Address

Action 12: IANA is requested to assign a TRILL multicast MAC address from the "TRILL Multicast Addresses" registry on the TRILL Parameters IANA web page [value 01-80-C2-00-00-47 recommended]. Description is "TRILL-ES-IS". Reference is [this document].

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>.

[RFC7177] - Eastlake 3rd, D., Perlman, R., Ghanwani, A., Yang, H., and V. Manral, "Transparent Interconnection of Lots of Links

(TRILL): Adjacency", RFC 7177, DOI 10.17487/RFC7177, May 2014,

<http://www.rfc-editor.org/info/rfc7177>.

[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>.

[RFC7356] - Ginsberg, L., Previdi, S., and Y. Yang, "IS-IS Flooding Scope Link State PDUs (LSPs)", RFC 7356, DOI 10.17487/RFC7356, September 2014, <http://www.rfc-editor.org/info/rfc7356>.

[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>.

[RFC7780] - Eastlake 3rd, D., Zhang, M., Perlman, R., Banerjee, A., Ghanwani, A., and S. Gupta, "Transparent Interconnection of Lots of Links (TRILL): Clarifications, Corrections, and Updates", RFC 7780, DOI 10.17487/RFC7780, February 2016, <http://www.rfc-editor.org/info/rfc7780>.

[RFC7961] - Eastlake 3rd, D. and L. Yizhou, "Transparent Interconnection of Lots of Links (TRILL): Interface Addresses APPsub-TLV", RFC 7961, DOI 10.17487/RFC7961, August 2016, <http://www.rfc-editor.org/info/rfc7961>.

[rfc6439bis] - D. Eastlake, Y. Li, M. Umair, A. Banerjee, and F. Hu, "Routing Bridges (RBridges): Appointed Forwarders", draft-ietf-trill-rfc6439bis, 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.

[RFC7978] - Eastlake 3rd, D., Umair, M., and Y. Li, "Transparent Interconnection of Lots of Links (TRILL): RBridge Channel Header Extension", RFC 7978, DOI 10.17487/RFC7978, September 2016, <http://www.rfc-editor.org/info/rfc7978>.

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

[DirAsstEncap] L. Dunbar, D. Eastlake, R. Perlman, I. Gashingksy, "Directory Assisted TRILL Encapsulation", draft-ietf-trill-directory-assisted-encap, work in progress.

[SmartEN] R. Perlman, F. Hu, D. Eastlake, K. Krupakaran, T. Liao, "TRILL Smart Endnodes", draft-ietf-trill-smart-endnodes", draft-ietf-trill-smart-endnodes, work in progress.

[X.233] - ITU-T Recommendation X.233: Protocol for providing the connectionless-mode network service: Protocol specification, International Telecommunications Union, August 1997

Acknowledgments

The contributions of the following persons are gratefully acknowledged:

Amanda Barber, Matthew Bocci, Alissa Cooper, Stephen Farrell, Daniel Franke, Igor Gashinski, Joel Halpern, Susan Hares, Alexey Melnikov, Gsyle Noble, Tianran Zhou

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

Yizhou Li
Huawei Technologies
101 Software Avenue,
Nanjing 210012, China

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

Copyright, Disclaimer, and Additional IPR Provisions