INTERNET-DRAFT
Intended status: Proposed Standard
Updates: ESADI

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

Expires: August 13, 2014
February 14, 2014

TRILL: Edge Directory Assist Mechanisms
<draft-ietf-trill-directory-assist-mechanisms-00.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.

Status of This Memo

Table of Contents

1. Introduction

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

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

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

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

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

1.1 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:

COP: Complete Push flag bit. See Sections 2 and 8.1 below.

CSNP Time: Complete Sequence Number PDU Time. See ESDADI [RFCesadi] and Section 8.1 below.

Data Label: VLAN or FGL.

FGL: Fine Grained Label [RFCfgl].

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

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

PSH: Push Directory flag bit. See Sections 2 and 8.1 below.

PUL: Pull Directory flag bit. See Sections 3 and 8.3 below.

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

RBridge: An alternative name for a TRILL switch.

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

tenant: Sometimes used as a synonym for FGL.

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 interface and the TRILL switches from which those interfaces are reachable [IA]. This service is scoped by Data Label (VLAN or FGL [RFCfgl]). A Push Directory also advertises whether or not it believes it has pushed complete mapping information for a Data Label. It might be pushing only a subset of the mapping and/or reachability information for a Data Label. The Push Model uses the ESADI [RFCesadi] protocol as its distribution mechanism.

With the Push Model, if complete address mapping information for a Data Label being pushed is available, a TRILL switch (RBridge) which has that complete pushed 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 request information from the directory server(s). TRILL switches simply use the ESADI [RFCesadi] 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 [RFC6326bis]. This will cause them to be pushed the Directory information for all such Data Labels that are being served by one or more Push Directory servers.

2.2 Push Directory Servers

Push Directory servers advertise their availability to push the mapping information for a particular Data Label to each other and to ESADI participants for that Data Label through ESADI by turning on the a flag bit in their ESADI Parameter APPsub-TLV for that ESADI instance (see [RFCesadi] and Section 8.1). Each Push Directory server MUST participate in ESADI for the Data Labels for which it will push mappings and set the PSH (Push Directory) bit in its ESADI-Parameters APPsub-TLV for that Data Label.

For robustness, it is useful to have more than one copy of the data being pushed. Each Push Directory server is configured with a number

in the range 1 to 8, which defaults to 2, for each Data Label for which it can push directory information. If the Push Directories for a Data Label are configured the same in this regard and enough such servers are available, this is the number of copies of the directory that will be pushed.

Each Push Directory server also has an 8-bit priority to be Active (see Section 8.1 of this document). This priority is treated as an unsigned integer where larger magnitude means higher priority and is in its ESADI Parameter APPsub-TLV. In cases of equal priority, the 6-byte IS-IS System IDs of the tied Push Directories are used as a tie breaker and treated as an unsigned integer where larger magnitude means higher priority.

For each Data Label it can serve, each Push Directory server orders, by priority, the Push Directory servers that it can see in the ESADI link state database for that Data Label that are data reachable [RFCclear] and determines its own position in that order. If a Push Directory server is configured to believe that N copies of the mappings for a Data Label should be pushed and finds that it is number K in the priority ordering (where number 1 is highest priority and number K is lowest), then if K is less than or equal to N the Push Directory server is Active. If K is greater than N it is Passive. Active and Passive behavior are specified below.

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

2.3 Push Directory Server State Machine

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

2.3.1 Push Directory States

A Push Directory Server is in one of six states, as listed below, for each Data Label it can serve. In addition, it has an internal State-Transition-Time variable for each Data Label it can serve which is set at each state transition and which enables it to determine how long it has been in its current state for that Data Label.

Down: 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 [RFCesadi] with the PSH bit 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.

Passive: 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 [RFCesadi] and advertises its ESADI fragment zero that includes an ESADI-Parameters APPsub-TLV with the PSH bit set to one and COP (Complete Push) bit zero.

Active: If a Push Directory server is Active, it advertises its directory data and any changes through ESADI [RFCesadi] in its ESADI-LSPs using the Interface Addresses [IA] APPsub-TLV and updates that information as it changes. The PSH bit is set to one in the ESADI-Parameters and the COP bit set to zero.

Completing: Same behavior as the Active state but responds differently to events.

Complete: The same behavior as Active except that the COP bit in the ESADI-Parameters APPsub-TLV is set to one and the server responds differently to events.

Reducing: The same behavior as Complete but responds differently to events. The PSH bit remains a one but the COP bit is cleared to zero in the ESADI-Parameters APPsub-TLV. Directory updates continue to be advertised.

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: The Push Directory server determines that it is priority K among the data reachable Push Directory servers (where highest priority is 1), the server is configured that there should be N copies pushed, and K is less than or equal to N. For example, the Push Directory server is configured that 2 copies should be pushed and finds that it is priority 1 or 2 among the Push Directory servers it can see.

The Pacify Condition: The Push Directory server determines that it is priority K among the data reachable data reachable Push Directory servers (where highest priority is 1), the server is configured that there should be N copies pushed, and K is greater than N. For example, the Push Directory server is configured that 2 copies should be pushed and finds that it is priority 3 or lower priority (higher number) among the Push directory servers it can see.

The Time Condition: The Push Directory server has been in its current state for an amount of time equal to or larger than its CSNP time (see Section 8.1).)

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

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

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

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

4. The server determines that the Pacify Condition is met.

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

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

7. The Time Condition is met.

2.3.3 State Transition Diagram and Table

The state transition table is as follows:

Event	Down	Passive	Active	Completing	Complete	Reducing
1	Passive	Passive	Active	Completing	Complete	Reducing
2	Down	Down	Passive	Passive	Reducing	Reducing
3	Down	Active	Active	Active	Reducing	Reducing
4	Down	Passive	Passive	Passive	Reducing	Reducing
5	Down	Completing	Complete	Completing	Complete	Complete
6	Down	Passive	Active	Active	Reducing	Reducing
7	Down	Passive	Active	Complete	Complete	Active

The above state table is equivalent to the following transition

diagram:

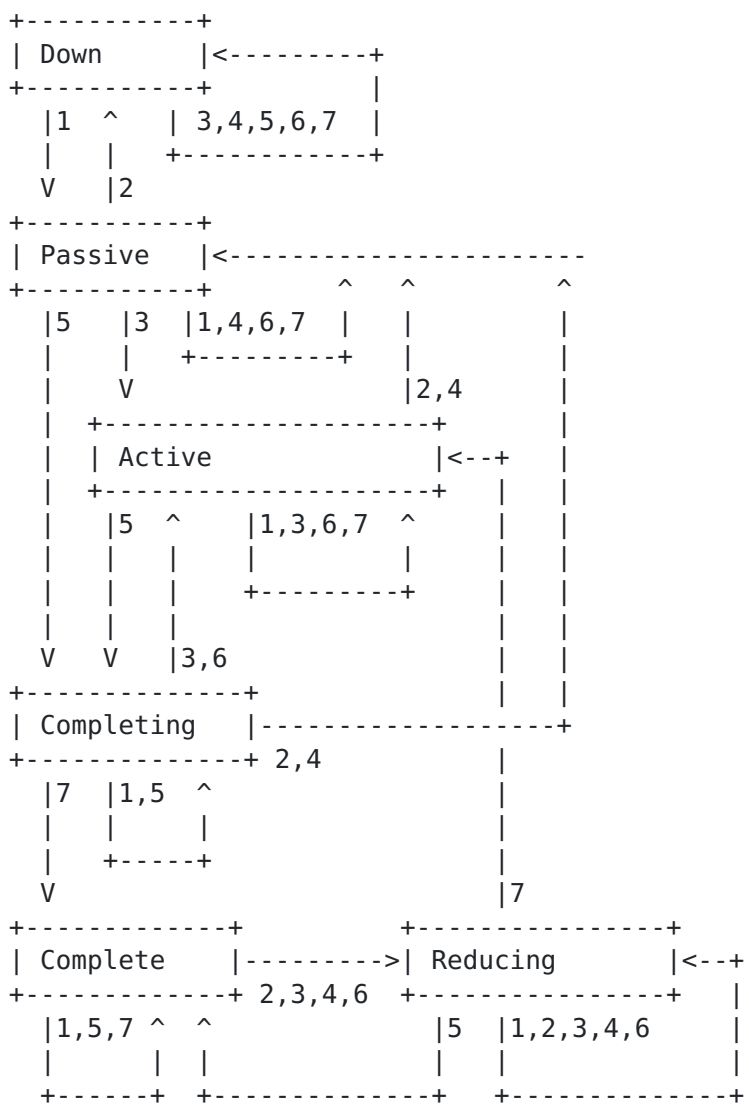


Figure 1. Push Server State Diagram

2.4 Additional Push Details

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

TRILL switches, whether or not they are a Push Directory server, MAY continue to advertise any locally learned MAC attachment information in ESDADI [RFCesadi] 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 may need to advertise interest in Data Labels even if it does not want to receive end station multidestination data in those Data Labels, the No Data (NOD) flag bit is provided as specified in Section 8.3.

When a Push Directory server is no longer data reachable [RFCclear], 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 PSH and COP bits are on in the ESADI zero fragment from the server for the relevant Data Label.
2. It has had continuous data connectivity to the server for the larger of the client's and the server's CSNP times.

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 have not yet received all of the ESADI LSP fragment from the Push Directory server.

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

2.5 Primary to Secondary Server Push Service

A secondary Push or Pull Directory server is one that obtains its data from a primary directory server. Other techniques MAY be used but, by default, this data transfer occurs through the primary server acting as a Push Directory server for the Data Labels involved while the secondary directory server takes the pushed data it receives from the highest priority Push Directory server and re-originates it. Such

a secondary server may be a Push Directory server or a Pull Directory server or both for any particular Data Label.

3. Pull Model Directory Assistance Mechanisms

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

Pull Directory servers for a particular Data Label X are found by looking in the core TRILL IS-IS link state database for data reachable TRILL switches that advertise themselves by having the Pull Directory flag (PUL) on in their Interested VLANs or Interested Labels sub-TLV [RFC6326bis] for that Data Label. If multiple such TRILL switches indicate that they are Pull Directory Servers for a particular Data Label, pull requests can be sent to any one or more of them but it is RECOMMENDED that pull requests be preferentially sent to the server or servers that are lower cost from the requesting TRILL switch.

Pull Directory requests are sent by enclosing them in an RBridge Channel [Channel] message using the Pull Directory channel protocol number (see Section 8.2). 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 can sent unsolicited Update messages to client TRILL switches that 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 returns can be sent for queries or updates as described in Section 3.5.

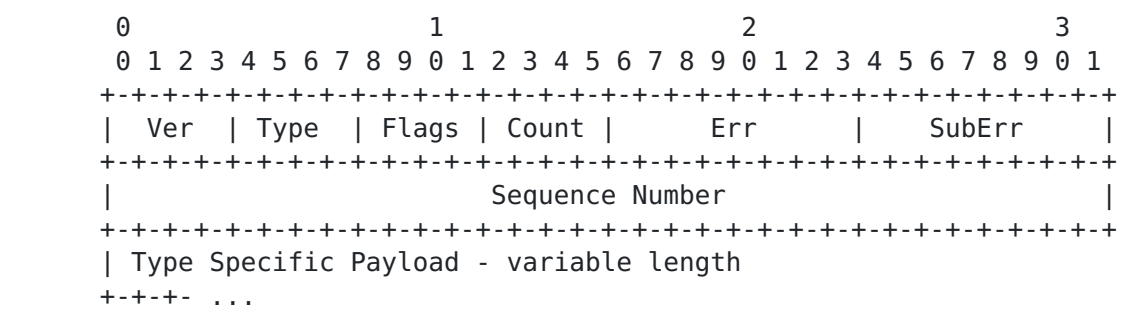
The requests to Pull Directory Servers are typically derived from ingressed ARP [RFC826], ND [RFC4861], or RARP [RFC903] messages, or data frames with unknown unicast destination MAC addresses, intercepted by an ingress TRILL switch as described in Section 4.

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. Thus both positive responses with data and negative responses can be cached and used to locally handle ARP, ND, RARP, or unknown destination MAC frames, until the responses expire. If information previously pulled is about to expire, a TRILL switch MAY try to refresh it by issuing a new pull request but, to avoid unnecessary requests, SHOULD NOT do so if it has not been recently used. The validity timer of cached Pull Directory responses is NOT reset or extended merely because that cache entry is used.

3.1 Pull Directory Message Common Format

All Pull Directory messages are transmitted as the payload of RBridge Channel messages. All Pull Directory messages are formatted as

described below starting with the following common 8-byte header:



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

Type: The Pull Directory message type as follows:

Type	Section	Name
0	3.2.1	Query
1	3.2.2	Response
2	3.1.4	Update
3	3.1.5	Acknowledge
4-15	-	Reserved

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

Count: Most 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 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 or acknowledgements. In messages that are requests or updates, these fields MUST be sent as zero and ignored on receipt. The meaning of values in the Err field depends on the Pull Directory message Type but in all cases the value zero means no error. The meaning of values in the SubErr field depends on both the message Type and 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 opaque 32-bit quantity set by the TRILL switch sending a request or other unsolicited message and returned in any reply or acknowledgement. It is used to match up responses

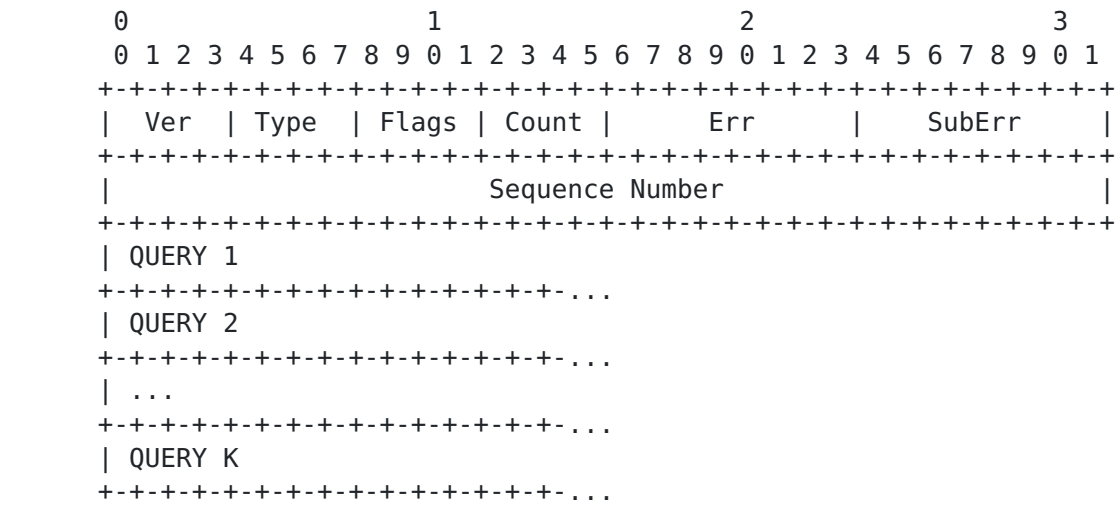
with the message to which they respond.

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

3.2 Pull Directory Query and Response Messages

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 [Channel] TRILL Data packet or as a native RBridge Channel data frame (see Section 3.4). The Data Label of the packet is the Data Label in which the query is being made. The priority of the channel message is a mapping of the priority of the frame being ingressed that caused the query with the default mapping depending, per Data Label, on the strategy (see Section 6) or a configured priority for generated queries. The Channel Protocol specific data is formatted as a header and a sequence of zero or more QUERY Records as follows:



Ver, Sequence Number: See 3.1.

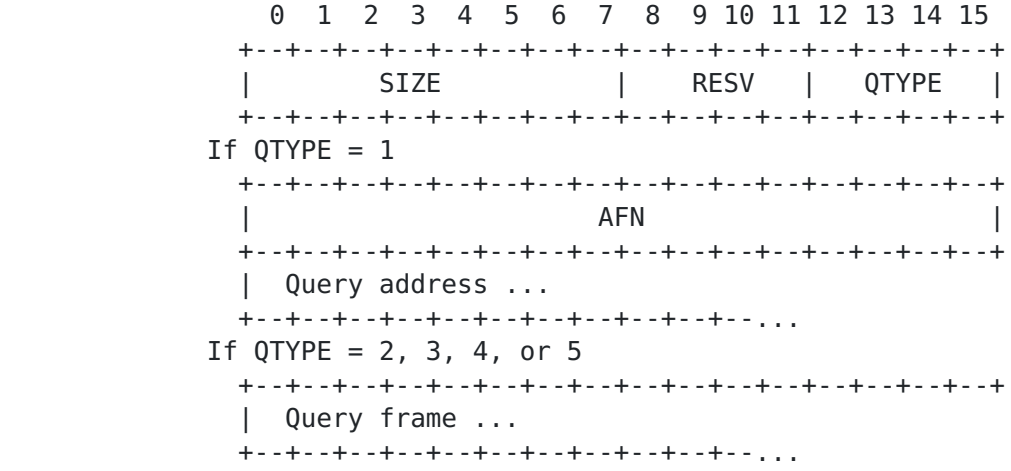
Type: 1 for Query. Queries received by an TRILL switch that is not a Pull Directory result in an error response (see Section 3.5) unless inhibited by rate limiting.

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

Count: Number of QUERY Records present. A Query message Count of

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

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



SIZE: Size of the QUERY record in bytes as an unsigned integer starting after the SIZE field and following byte. Thus the minimum legal value is 2. A value of SIZE less than 2 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.

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

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

QTYPE	Description
0	reserved
1	address query
2	ARP query frame
3	ND query frame
4	RARP query frame
5	Unknown unicast MAC query frame
6-14	assignable by IETF Review
15	reserved

AFN: Address Family Number of the query address.

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

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

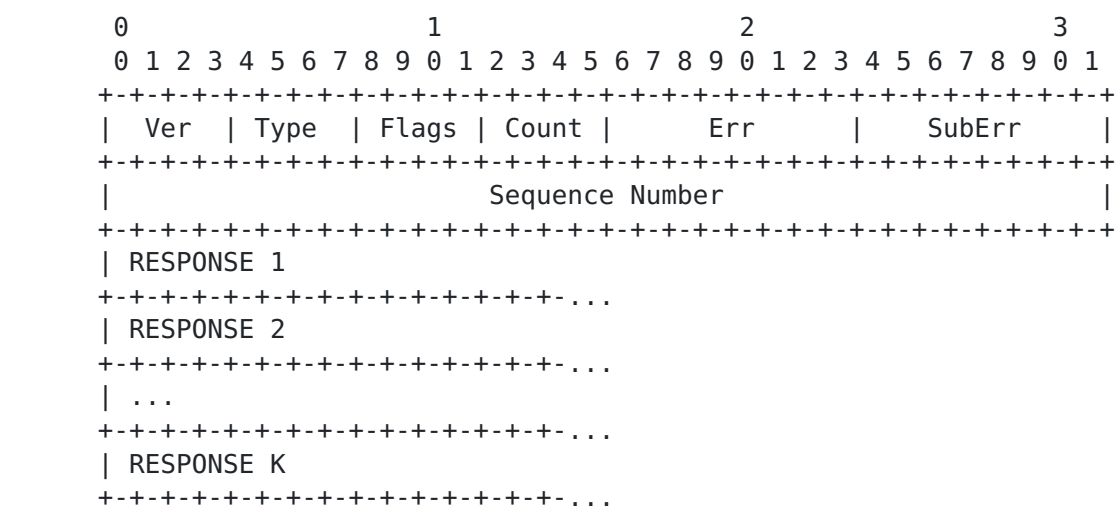
If no response is received to a Pull Directory Query message within a timeout configurable in milliseconds that defaults to 200, the Query message should be re-transmitted with the same Sequence Number up to a configurable number of times that defaults to three. If there are multiple QUERY Records in a Query message, responses can be received to various subsets of these QUERY Records before the timeout. In that case, the remaining unanswered QUERY Records should be re-sent in a new Query message with a new sequence number. If a TRILL switch is not capable of handling partial responses to queries with multiple QUERY Records, it MUST NOT sent a Request message with more than one QUERY Record in it.

See Section 3.5 for a discussion of how Query message errors are handled.

3.2.2 Pull Directory Response Format

Pull Directory Response messages are sent as the Channel Protocol specific content of an RBridge Channel message [Channel] TRILL Data packet or as a native RBridge Channel data frame (see Section 3.4). Responses are sent with the same Data Label and priority as the Query message to which they correspond except that the Response message priority is limited to be not more than a configured value. This priority limit is configurable at per TRILL switch and defaults to priority 6. Pull Directory Response messages SHOULD NOT be sent with priority 7 as that priority SHOULD be reserved for messages critical to network connectivity.

The RBridge Channel protocol specific data format is as follows:



Ver, Sequence Number: As specified in Section 3.1.

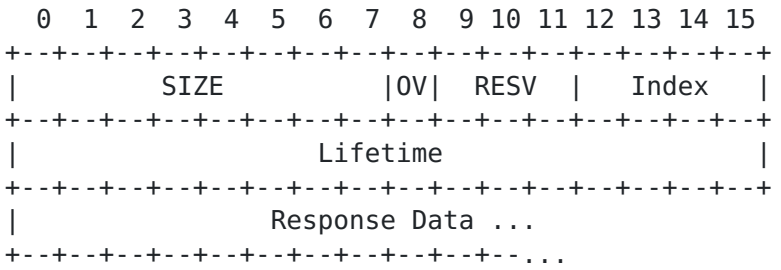
Type: 2 = Response.

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 3.5.

RESPONSE: Each RESPONSE record within a Pull Directory Response message is formatted as follows:



SIZE: Size of the RESPONSE Record in bytes starting after the SIZE field and following byte. Thus the minimum value of SIZE is 2. If SIZE is less than 2, that RESPONSE Record and all subsequent RESPONSE Records in the Response message 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: Four 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 200 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 3.6 hours.

Response Data: There are various types of RESPONSE Records.

- If the Err field is non-zero, then the Response Data is a copy of the corresponding QUERY Record data, that is, either an AFN followed by an address or a query frame. See Section 3.5 for additional information on errors.
- If the Err field is zero and the corresponding QUERY Record was an address query, then the Response Data is the contents of an Interface Addresses APPsub-TLV [IA]. The maximum size of such contents is 253 bytes in the case when SIZE is 255.
- If the Err field is zero and the corresponding QUERY Record was a frame query, then the Response data consists of the response frame for ARP, ND, or RARP and a copy of the frame for unknown unicast destination MAC.

Multiple RESPONSE Records can appear in a Response message with the same index if the answer to a QUERY Record consists of multiple Interface Address APPsub-TLV contents. 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 3.5 for a discussion of how errors are handled.

3.3 Cache Consistency

A Pull Directory MUST take action to minimize the amount of time that a TRILL switch will continue to use stale information from that Pull Directory by sending Update messages.

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

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

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

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

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

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

If method 1, the most crude 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 Update message is flooded within that Data Label as an RBridge Channel message with an Inner.MacDA of All-Egress-RBridges. And if data is added and there are outstanding cached negative responses, an all-addresses flush negative message is similarly flooded. "All-addresses" is indicated by the Count field being zero in an Update message. 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 all cached data responses it has for that Data Label. Similarly, on receiving an all addresses flush negative Update, indicated by the F and N bits being one and the Count being zero, it discards all cached negative replies for that Data Label. A combined flush positive and negative can be flooded by having all of the F, P, and N bits set to one resulting in the discard of all positive and negative cached information for the Data Label.

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 similar to the method 1 flooded flush Update messages and are also sent as RBridge Channel messages with an Inner.MacDA of All-Egress-RBridges. However the Count field will be non-zero and either the P or N bit, but not both, will be one. On receiving such as address specific unsolicited update, if it is positive the addresses in the RESPONSE records in the unsolicited response are compared to the addresses about which the receiving TRILL switch is holding cached positive information from that server and, if they match, the cached information is updated. On receiving an address specific unsolicited update negative message, the addresses in the RESPONSE records in the unsolicited update are compared to the addresses about which the receiving TRILL switch is holding cached negative information from that server and, if they match, the cached negative information is updated.

If method 3 is being followed, the same sort of unsolicited update messages are sent as with method 2 above except they are not normally flooded but unicast only to the specific TRILL switches the directory

server believes may be holding the cached positive or negative information that needs updating. However, a Pull Directory server MAY flood the unsolicited update under method 3, for example if it determines that a sufficiently large fraction of the TRILL switches in some Data label are requesters that need to be updated.

A Pull Directory server tracking cached information with method 3 MUST NOT clear the indication that it needs update cached information at a querying TRILL switch until it has sent an Update message and received a corresponding Acknowledge message or it has sent a configurable number of updates at a configurable interval which default to 3 updates 200 milliseconds apart.

A Pull Directory server tracking cached information with methods 2 or 1 SHOULD NOT clear the indication that it needs to update cached information until it has sent an Update message and received a corresponding Acknowledge message from all of its ESADI neighbors or it has sent a configurable number of updates at a configurable interval that defaults to 3 updates 200 milliseconds apart.

3.3.1 Update Message Format

An Update message is formatted as a Response message except that the Type field in the message header is a different value.

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

The Flags field of the message header for an Update message is as follows:

F: The Flood bit. If zero, the response is to be 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

3.3.2 Acknowledge Message Format

An Acknowledge message is sent in response to an Update to confirm receipt or indicate an error unless response is inhibited by rate limiting. It is also formatted as a Response message.

If there are no errors in the processing of an Update message, the message is essentially echoed back with the Type changed to Acknowledge.

If there was an overall or header error in an Update message, it is echoed back as an Acknowledge message with the Err and SubErr fields set appropriately (see Section 3.5).

If there is a RESPONSE Record level error in an Update message, one or more Acknowledge messages may be returns as indicated in Section 3.5.

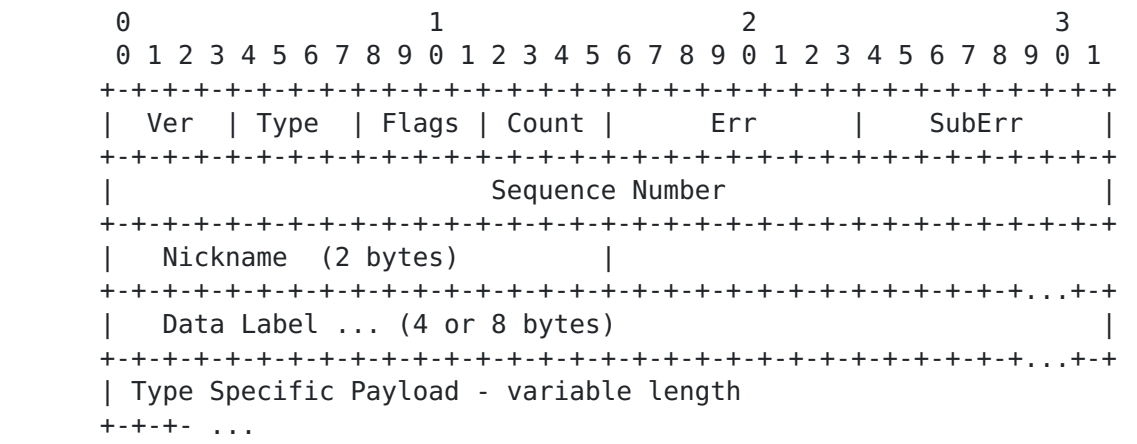
3.4 Pull Directory Hosted on an End Station

Optionally, a Pull Directory actually hosted on an end station MAY be supported. In that case, a TRILL switch must proxy for the end station and advertise itself as a Pull Directory server.

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

The native Pull Directory RBridge Channel messages use the same Channel protocol number as do the inter-RBridge Pull Directory RBridge Channel messages. The native messages SHOULD be sent with an Outer.VLAN tag which gives the priority of each message which is the priority of the original inter-RBridge request packet. The Outer.VLAN ID used is the Designated VLAN on the link to the end station. Since there is no TRILL Header or inner Data Label for native RBridge Chanel messages, that information is added to the header.

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



Fields not described below are as in Section 3.1.

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

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

3.5 Pull Directory Message Errors

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

If there is an error that applies to an entire Query message or its header, as indicated by the range of the value of the Err field, then the QUERY records in the request are just echoed back in the RESPONSE records of the Response message but expanded with a zero Lifetime and the insertion of the Index field. If there is an error that applies

to an entire Update message or its header, then the RESPONSE records in the update, if any, are echoed back in the Acknowledge 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 Response message. In an error Response message, the QUERY Record or records being responded to appear, expanded by the Lifetime for which the server thinks the error might persist and with their Index inserted, as the RESPONSE record or records.

If errors occur at the RESPONSE Record level for an Update message, they MUST be reported in a Acknowledge message separate from the acknowledgement of any non-erroneous RESPONSE Records. If multiple RESPONSE Records in an Update have different errors, they MUST be reported in separate Acknowledge messages. If multiple RESPONSE Records in an Update message have the same error, this error response MAY be reported in one Acknowledge message. 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 127 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.

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

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

SubErr	Field with Error
0	Unspecified
1	Unknown V field value
2	Reserved T field value
3	Zero sequence number in request
4-254	(Available for allocation by Expert Review)
255	Reserved

The following sub-errors are specified under error code 128 and 130:

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

More TBD

3.6 Additional Pull Details

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

Because a Pull Directory server may need to advertise interest in Data Labels even though it does not want to received end station data in those Data Labels, the No Data (NOD) flag bit is provided as specified in Section 8.3. For example, an RBridge hosting a Pull Directory may be a secondary directory that wants to receive its data from a primary Push Directory server but have no interest in receiving multicast traffic from end stations.

4. Events That May Cause Directory Use

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.

ARP requests and replies normally have the broadcast address in their MAC destination address and are normally treated the same way as any broadcast Ethernet frame. A directory assisted RBridge MUST intercept ARP broadcast, ND multicast, and unknown unicast destination MAC address native frames. It SHOULD also intercept RARP and, if complete directory information is available, forged source MAC frames.

Support for each of the cases below is separately optional.

4.1 Forged Native Frame Ingress

End stations can forge the source MAC and/or IP address in a native frame that an edge TRILL switch receives for ingress in some particular Data Label. If there is complete Directory information as to what end stations should be reachable by an egress TRILL switch, frames with forged source addresses SHOULD be discarded. If such frames are discarded, then none of the special processing in the remaining subsection of this Section 2 occur and MAC address learning (see [RFC6325] Section 4.8) SHOULD NOT occur. ("SHOULD NOT" is chosen because it is harmless in cases where it has no effect. For example, if complete directory information is available and such directory information is treated as having a higher confidence that MAC addresses learned from the data plane.)

If directory information includes the TRILL switch a port by which a MAC and/or IP address is reachable, that may also be tested on ingress so that an end station on one TRILL switch port cannot forge a source MAC or IP address that should not be reachable by that port even if it is reachable by that TRILL switch.

4.2 Unknown Destination MAC

Ingressing a native frame with an unknown unicast destination MAC:

The mapping from the destination MAC and Data Label to the egress TRILL switch from which it is reachable is needed to ingress the frame as unicast. If the egress TRILL switch is unknown, the frame

must be either dropped or ingressed as a multi-destination frame which is flooded to all edge TRILL switches for its Data Label resulting in increased link utilization compared with unicast routing. Depending on the configuration of the TRILL switch ingressing the native frame (see Section 6), directory information can be used for the { destination MAC, Data Label } to egress TRILL switch nickname mapping and destination MACs for which such direction information is not available MAY be discarded.

4.3 Address Resolution Protocol (ARP)

Ingressing an ARP [RFC826]:

ARP is a flexible protocol detected by its Ethertype of 0x0806. It is commonly used on a link to (1) query for the MAC address corresponding to an IPv4 address, (2) test if an IPv4 address is in use, or (3) to announce a change in any of IPv4 address, MAC address, and/or point of attachment.

The logically important elements in an ARP are (1) the specification of a "protocol" and a "hardware" address type, (2) an operation code that can be Request or Reply, and (3) fields for the protocol and hardware address of the sender and the target (destination) node.

Examining the three types of ARP use:

1. General ARP Request / Response

   This is a request for the destination "hardware" address corresponding to the destination "protocol" address; however, if the source and destination protocol addresses are equal, it should be handled as in type 2 below. A general ARP is handled by doing a directory lookup on the destination "protocol" address provided in hops of finding a mapping to the desired "hardware" address. If such information is obtain from a directory, a response can be synthesized.

2. Address Probe ARP Query

   An address probe ARP is used to determine if an IPv4 address is in use [RFC5227]. It can be identified by the source "protocol" (IPv4) address field being zero. The destination "protocol" address field is the IPv4 address being tested. If some host believes it has that destination IPv4 address, it would respond to the ARP query, which indicates that the address is in use. Address probe ARPs can be handled in the same way as General ARP queries above.

3. Gratuitous ARP

A gratuitous ARP is an unsolicited ARP message, usually a response but sometimes a query, used by a host to announce a new IPv4 address, new MAC address, and/or new point of network attachment. Such ARPs are identifiable because the sender and destination "protocol" address fields have the same value. Thus, under normal circumstances, there really isn't any separate destination host to generate a response. If complete Push Directory information is being used with the Notify flag set in the IA APPsub-TLVs being pushed [IA] by all the TRILL switches in the Data Label, then gratuitous ARPs SHOULD be discarded rather the ingressed. Otherwise, they are either ingressed and flooded or discarded depending on local policy.

4.4 IPv6 Neighbor Discovery (ND)

Ingressing an IPv6 ND [RFC4861]:
TBD

Secure Neighbor Discovery messages [RFC3971] will, in general, have to be sent to the neighbor intended so that neighbor can sign the answer; however, directory information can be used to unicast a Secure Neighbor Discovery packet rather than multicasting it.

4.5 Reverse Address Resolution Protocol (RARP)

Ingressing a RARP [RFC903]:
RARP uses the same packet format as ARP but a different Ethertype (0x8035) and opcode values. Its use is similar to the General ARP Request/Response as described above. The difference is that it is intended to query for the destination "protocol" address corresponding to the destination "hardware" address provided. It is handled by doing a directory lookup on the destination "hardware" address provided in hops of finding a mapping to the desired "protocol" address. For example, looking up a MAC address to find the corresponding IP address.

5. Layer 3 Address Learning

TRILL switches MAY learn IP addresses in a manner similar to that in which they learn MAC addresses. On ingress of a native IP frame, they can learn the { IP address, MAC address, Data Label, input port } set and on the egress of a native IP frame, they can learn the { IP address, MAC address, Data Label, remote RBridge } information plus the nickname of the RBridge that ingressed the frame.

This locally learned information is retained and times out in a similar manner to MAC address learning specified in [RFC6325]. By default, it has the same Confidence as locally learned MAC reachability information.

Such learned Layer 3 address information MAY be disseminated with ESDADI [RFCesadi] using the IA APPsub-TLV [IA]. It can also be used as, in effect, local directory information to assist in locally responding to ARP/ND packets as discussed in Section 4.

6. 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 a majority of Data Labels, if not all.

For those edge TRILL switch nodes, a hybrid model should be considered. That is the Push Model is used for some Data Labels, and the Pull Model is used for other Data Labels. 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 which only communicate with external peers occasionally for management interface, the mapping entries for those VLANs should be pulled down from directory when the need comes up.

The mechanisms described above for Push and Pull Directory services make it easy to use Push for some Data Labels 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.

For Data Labels in which a hybrid push/pull approach is being taken, it would make sense to use push for address information of hosts that frequently communicate with many other hosts in the Data Label, such as a file or DNS server. Pull could then be used for hosts that communicate with few other hosts, perhaps such as hosts being used as compute engines.

6.1 Strategy Configuration

Each TRILL switch that has the ability to use directory assistance has, for each Data Label X in which it is might ingress native frames, one of four major modes:

0. No directory use: The TRILL switch does not subscribe to Push Directory data or make Pull Directory requests for Data Label X and directory data is not consulted on ingressed frames in Data Label X that might have used directory data. This includes ARP,

ND, RARP, and unknown MAC destination addresses, which are flooded as appropriate.

1. Use Push only: The TRILL switch subscribes to Push Directory data for Data Label X.

2. Use Pull only: When the TRILL switch ingresses a frame in Data Label X that can use Directory information, if it has cached information for the address it uses it. If it does not have either cached positive or negative information for the address, it sends a Pull Directory query.

3. Use Push and Pull: The TRILL switch subscribes to Push Directory data for Data Label X. When it ingresses a frame in Data Label X that can use Directory information and it does not find that information in its link state database of Push Directory information, it makes a Pull Directory query.

The above major Directory use mode is per Data Label. In addition, there is a per Data Label per priority minor mode as listed below that indicates what should be done if Directory Data is not available for the ingressed frame. In all cases, if you are holding Push Directory or Pull Directory information to handle the frame given the major mode, the directory information is simply used and, in that instance, the minor mode does not matter.

A. Flood immediate: Flood the frame immediately (even if you are also sending a Pull Directory) request.

B. Flood: Flood the frame immediately unless you are going to do a Pull Directory request, in which case you wait for the response or for the request to time out after retries and flood the frame if the request times out.

C. Discard if complete or Flood immediate: If you have complete Push Directory information and the address is not in that information, discard the frame. If you do not have complete Push Directory information, the same as A above.

D. Discard if complete or Flood: If you have complete Push Directory information and the address is not in that information, discard the frame. If you do not have complete Push Directory information, the same as B above.

In addition, the query message priority for Pull Directory requests sent can be configured on a per Data Label, per ingressed frame priority basis. The default mappings are as follows where Ingress Priority is the priority of the native frame that provoked the Pull Directory query:

Ingress Priority	If Flood Immediate	If Flood Delayed
7	5	6
6	5	6
5	4	5
4	3	4
3	2	3
2	0	2
0	1	0
1	1	1

Priority 7 is normally only used for urgent messages critical to adjacency and so is avoided by default for directory traffic. Unsolicited updates are sent with a priority that is configured per Data Label that defaults to priority 5.

7. Security Considerations

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

- Incorrect directory mappings can result in data being delivered to the wrong end stations, or set of end stations in the case of multi-destination packets, violation security policy.

- Missing or incorrect directory data can result in denial of service due to sending data packets to black holes or discarding data on ingress due to incorrect information that their destinations are not reachable.

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

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

For general TRILL security considerations, see [RFC6325].

8. IANA Considerations

This section gives IANA allocation and registry considerations.

8.1 ESADI-Parameter Data Extensions

IANA is requested to allocate two ESADI-Parameter TRILL APPsub-TLV flag bits for "Push Directory" (PSH) and "Complete Push" (COP) and to create a sub-registry in the TRILL Parameters Registry as follows:

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

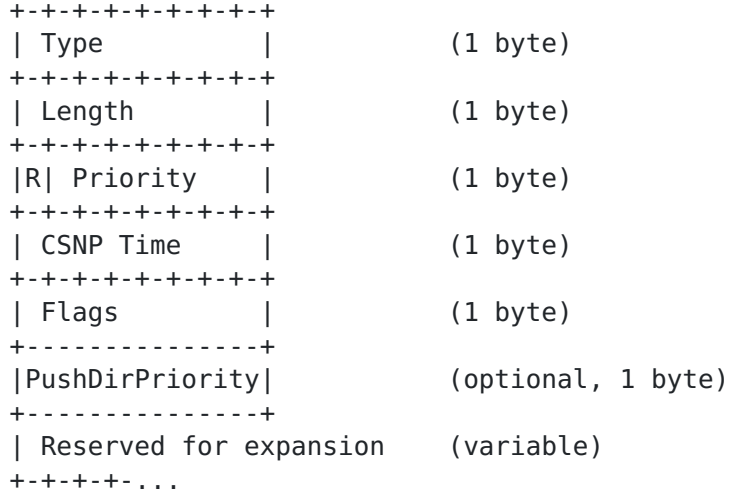
Registration Procedures: Expert Review

References: [RFCesadi] [This document]

Bit	Mnemonic	Description	Reference
0	UN	Supports Unicast ESADI	ESDADI [RFCesadi]
1	PSH	Push Directory Server	This document
2	COP	Complete Push	This document
3-7	-	available for allocation	

The COP bit is ignored if the PSH bit is zero.

In addition, the ESADI-Parameter APPsub-TLV is optionally extended, as provided in its original specification in ESDADI [RFCesadi], by one byte as show below:



The meanings of all the fields are as specified in ESDADI [RFCesadi] except that the added PushDirPriority is the priority of the advertising ESADI instance to be a Push Directory as described in

Section 2.3. If the PushDirPriority field is not present (Length = 3) it is treated as if it were 0x40. 0x40 is also the value used and placed here by an TRILL switch whose priority to be a Push Directory has not been configured.

8.2 RBridge Channel Protocol Number

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

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

IANA is requested to allocate two currently reserved bits in the Interested VLANs field of the Interested VLANs sub-TLV (suggested bits 18 and 19) and the Interested Labels field of the Interested Labels sub-TLV (suggested bits 6 and 7) [RFC6326bis] to indicate Pull Directory server (PUL) and No Data (NOD) respectively. These bits are to be added, with this document as reference, to the "Interested VLANs Flag Bits" and "Interested Labels Flag Bits" subregistries created by [RFCesadi].

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

An example of a TRILL switch serving as a directory that would not want multi-destination traffic in some Data Labels might be a TRILL switch that does not offer end station service for any of the Data Labels for which it is serving as a directory and is either a Pull Directory and/or a Push Directory for which all of the ESADI traffic can be handled by unicast ESDADI [RFCesadi].

Acknowledgments

The contributions of the following persons are gratefully acknowledged:

TBD

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

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.

[RFC6326bis] - Eastlake, D., Banerjee, A., Dutt, D., Perlman, R., and A. Ghanwani, "TRILL Use of IS-IS", draft-ietf-isis-rfc6326bis, work in progress.

[RFCclear] - Eastlake, D., M. Zhang, A. Ghanwani, V. Manral, A. Banerjee, draft-ietf-trill-clear-correct-06.txt, in RFC Editor's queue.

[Channel] - D. Eastlake, V. Manral, Y. Li, S. Aldrin, D. Ward, "TRILL: RBridge Channel Support", draft-ietf-trill-rbridge-channel-08.txt, in RFC Editor's queue.

[RFCfgl] - D. Eastlake, M. Zhang, P. Agarwal, R. Perlman, D. Dutt,

"TRILL: Fine-Grained Labeling", draft-ietf-trill-fine-labeling-07.txt, in RFC Editor's queue.

[RFCesadi] - Zhai, H., F. Hu, R. Perlman, D. Eastlake, O. Stokes, "TRILL (Transparent Interconnection of Lots of Links): The ESADI (End Station Address Distribution Information) Protocol", draft-ietf-trill-esadi, work in progress.

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

Informational References

[RFC5227] - Cheshire, S., "IPv4 Address Conflict Detection", RFC 5227, July 2008.

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

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

[ARP reduction] - Shah, et. al., "ARP Broadcast Reduction for Large Data Centers", Oct 2010.

Authors' Addresses

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

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

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

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

Radia Perlman
Intel Labs
2200 Mission College Blvd.
Santa Clara, CA 95054-1549 USA

Phone: +1-408-765-8080
Email: Radia@alum.mit.edu

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

Email: igor@yahoo-inc.com

Yizhou Li
Huawei Technologies
101 Software Avenue,
Nanjing 210012 China

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