INTERNET-DRAFT
Intended status: Proposed Standard
Updates: ESADI

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

Expires: April 20, 2014
October 21, 2013

TRILL: Edge Directory Assistance Mechanisms
<draft-dunbar-trill-scheme-for-directory-assist-06.txt>

Abstract

This document describes mechanisms for using directory server(s) to assist TRILL (Transparent Interconnection of Lots of Links) edge switches in reducing multi-destination traffic, particularly ARP/ND and unknown unicast flooding.

Status of This Memo

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

Distribution of this document is unlimited. Comments should be sent to the TRILL working group mailing list.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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

The list of current Internet-Drafts can be accessed at http://www.ietf.org/1id-abstracts.html. The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

Table of Contents

1. Introduction

[DirectoryFramework] describes a high-level framework 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, 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 [RFC5342bis] corresponds to an IP address within a Data Label (VLAN or FGL (Fine Grained Label [RFCfgl])) and from what egress TRILL switch (RBridge) (and optionally what specific TRILL switch port) 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. Modes of operation, including hybrid Push/Pull, are discussed in Section 4.

The mechanisms used to initially populate directory data in primary servers is beyond the scope of this document. The Push Directory service can be used by a primary server to provide Directory data to secondary servers as described in Section 2.

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 additions:

CP: Complete Push flag bit. See Sections 2 and 6.1 below.

CSNP Time: Complete Sequence Number PDU Time. See [ESADI] and Section 6.1 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.

PD: Push Directory flag bit. See Sections 2 and 6.1 below.

primary server: A Directory server that obtains the information it is serving up by a reliable mechanism outside the scope of this document but 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 [DirectoryFramework], one or more Push Directory servers 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 mapping information for only a subset of the ports in a Data Label. The Push Model uses the [ESADI] 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 can simply drop a native frame if the destination unicast MAC address can’t be found in the mapping information available, instead of flooding it (see Section 2.1). This will minimize flooding of packets due to errors or inconsistencies but is not practical if directories have incomplete information.

2.1 Requesting Push Service

In the Push Model, it is necessary to have a way for an RBridge to request information from the directory server(s). RBridges simply use the ESADI 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 by turning on a flag bit in their ESADI Parameter APPsub-TLV [ESADI] for that ESADI instance (see Section 8.1). Each Push Directory server MUST participate in ESADI for the Data Labels for which it will push mappings and set the PD (Push Directory) bit in their 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 RBridge that is a 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 ID is 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 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.

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 such Data Label which it set at each state transition and which enables it to determine how long it has been in its current state.

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 [ESADI] with the PD 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 [ESADI] and its [ESADI] fragment zero includes an ESADI-Parameters APPsub-TLV with the PD bit set to one and CP (Complete Push) bit zero.

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

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

Complete: The same behavior as Completing except that the CP 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 PD bit remains a one but the CP 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 server determines that there are K data reachable Push Directory servers, the server is configured that there should be N copies pushed, and K is less than or equal to N.

The Pacify Condition: The server determines that there are K data reachable Push Directory servers, the server is configured that there should be N copies pushed, and K is greater than N.

The Time Condition: The 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 or TRILL switch was configured to be down but the TRILL switch on which it resides is up and the server is configured to be up.

2. The Push Directory server or the TRILL switch on which it is resident 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 server is configured to believe it has complete data and the Activate Condition is met.

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 diagram is as follows.

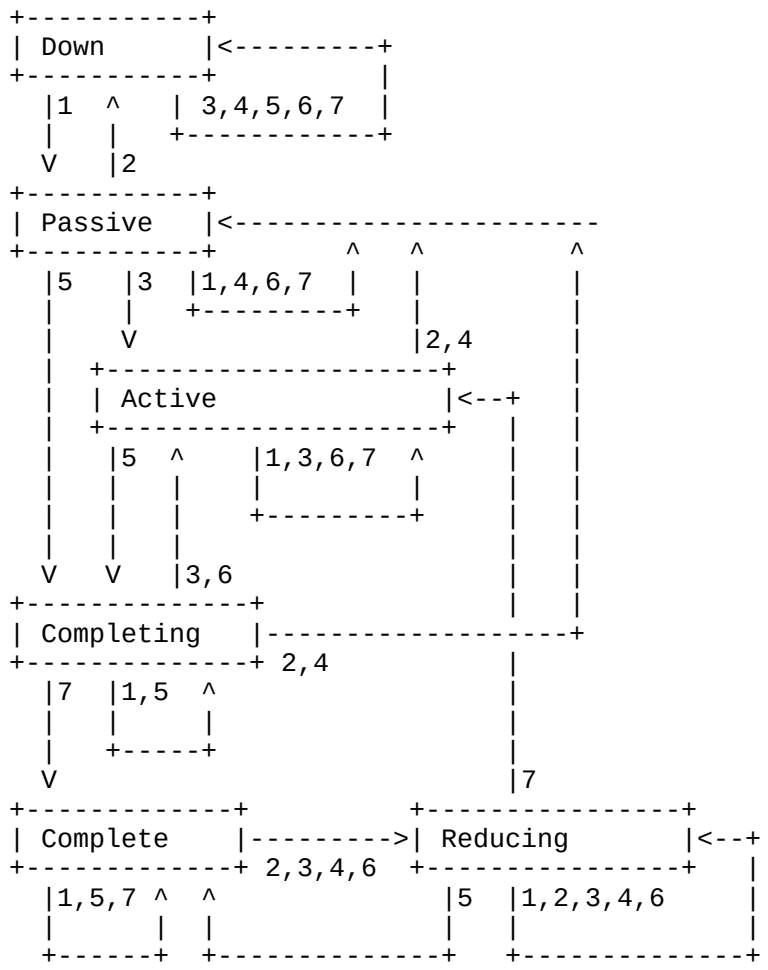


Figure 1. Push Server State Diagram

This state diagram is equivalent to the following transition table:

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

2.4 Additional Push Details

Push Directory mappings can be distinguished for any 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.

RBridges, whether or not they are a Push Directory server, MAY continue to advertise any locally learned MAC attachment information in [ESADI] 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 should generally not be done as it would not add anything and would just waste bandwidth and ESADI link state space. An exception would be when an RBridge 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 data in those Data Labels, the No Data flag bit is provided as discussed in Section 6.3.

If an RBridge notices that a Push Directory server is no longer data reachable [RFCclear], it 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 impractical for an RBridge receiving Push Directory information to ever 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 PD and CP bits are on in the ESADI zero fragment from the server for the relevant Data Label.
2. A client RBridge might be just coming up and receive an EASDI LSP meeting the requirement in point 1 above but have not yet received all of the ESADI LSP fragment from the Push Directory server. Thus, it should not believe that information to be complete unless it has also had data connectivity to the server for the larger of the client’s and the server’s CSNP times.

There may be transient 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 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 Push Directory server takes the pushed data it receives from the highest priority Push Directory server and re-originates it.

3. Pull Model Directory Assistance Mechanisms

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

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

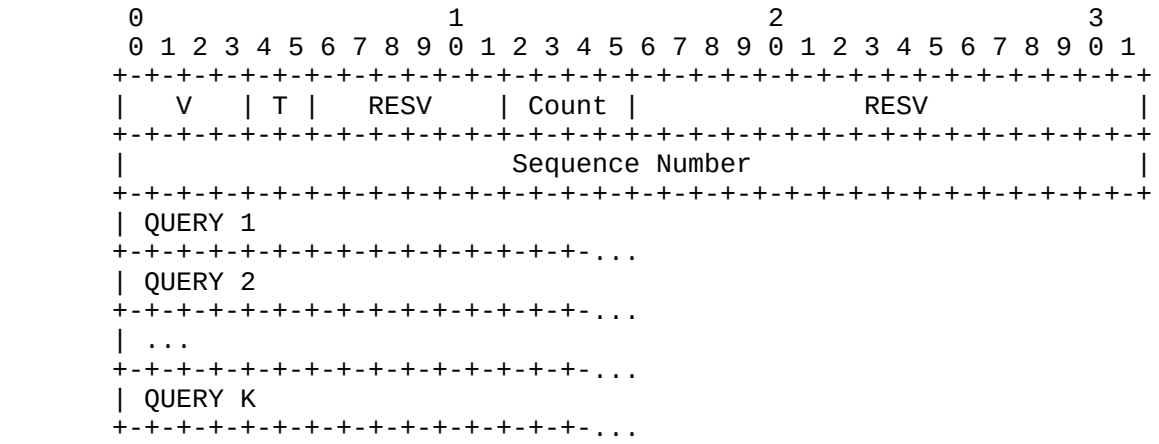
Pull Directory requests are sent by enclosing them in an RBridge Channel [Channel] message using the Pull Directory channel protocol number (see Section 8.2). Responses are returned in an RBridge Channel message using the same channel protocol number.

The requests to Pull Directory Servers are typically derived from normal ARP [RFC826], ND [RFC4861], RARP [RFC903] messages or data frames with unknown unicast destination MAC addresses intercepted by the RBridge 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 for immediate response to ARP, ND, RARP, or unknown destination MAC frames, until they expire. If information previously pulled is about to expire, an RBridge MAY try to refresh it by issued a new pull request but, to avoid unnecessary requests, SHOULD NOT do so if it has not been recently used.

3.1 Pull Directory Request Format

A Pull Directory request is sent as the Channel Protocol specific content of an inter-RBridge Channel message [Channel] TRILL Data packet. The Data Label in 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 request with the default mapping depending, per Data Label, on the strategy (see Section 6). The Channel Protocol specific data is formatted as follows:



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

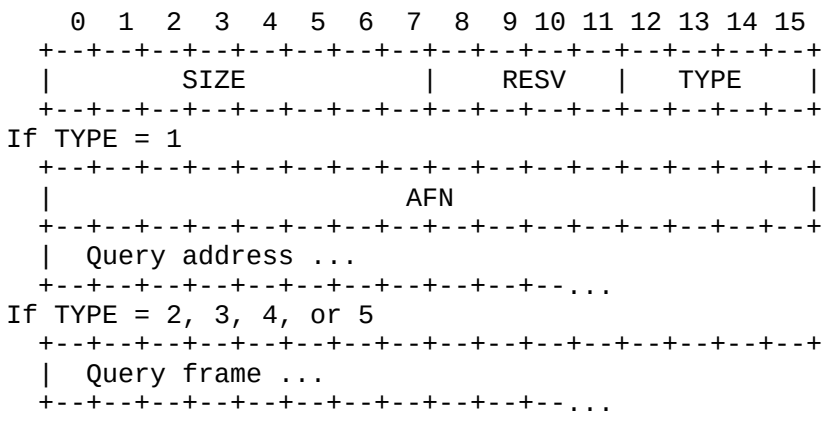
T: Type. 0 => Response, 1=> Query, 2=> Unsolicited Update, 3=> Reserved. An unsolicited update is formatted as a response except there is no corresponding query. Messages received with type = 3 are discarded. Queries received by an RBridge that is not a Pull Directory are discarded. Responses that do not match an outstanding Query are discarded.

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

Count: Number of queries present.

Sequence Number: A 32-bit quantity set by the sending RBridge, returned in any responses, and used to match up responses with queries. It is opaque except that the value zero is reserved for Unsolicited Update response messages. A Request received with Sequence Number zero is discarded.

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



SIZE: Size of the query data in bytes as an unsigned byte starting with and including the SIZE field itself. Thus the minimum legal value is 2. A value of SIZE less than 2 indicates a malformed message. The "QUERY" with the illegal SIZE value and all subsequent QUERYs 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.

TYPE: There are two types of queries currently defined, (1) a query that provides an explicit address and asks for other addresses for the interface specified by the query address and (2) a query that includes a frame. The fields of each are specified below. Values of TYPE are as follows

TYPE	Description
0	reserved
1	query address
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.

Query Address: The query is asking for any other addresses, and the RBridge 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 RBridge primarily interested in the RBridge by which that MAC address is reachable, or

(2) an IP address with the querying RBridge interested in the corresponding MAC address and the RBridge by which that MAC address is reachable. But it could be some other address type.

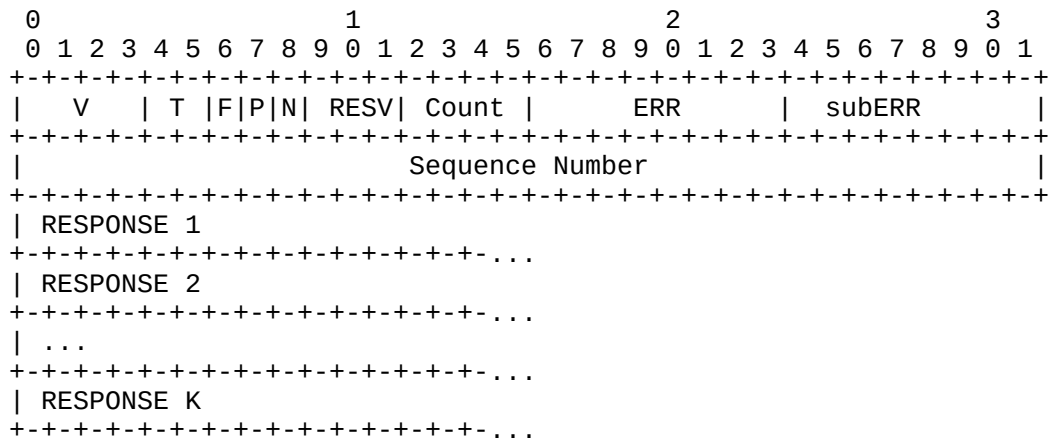
Query Frame: Where a Pull Directory query is the result of an ARP, ND, RARP, or unknown unicast MAC destination address, the ingress RBridge MAY send the frame to a Pull Directory Server if the frame is small enough to fit into a query message.

A query count of zero is explicitly allowed, for the purpose of pinging a Pull Directory server to see if it is responding to requests. On receipt of such an empty query message, a response message that also has a count of zero MUST be sent unless inhibited by rate limiting.

If no response is received to a Pull Directory request within a timeout configurable in milliseconds that defaults to 2,000, the request should be re-transmitted with the same Sequence Number up to a configurable number of times that defaults to three. If there are multiple queries in a request, responses can be received to various subsets of these queries by the timeout. In that case, the remaining unanswered queries should be re-sent in a new query with a new sequence number. If an RBridge is not capable of handling partial responses to requests with multiple queries, it MUST NOT sent a request with more than one query in it.

3.2 Pull Directory Response Format

Pull Directory responses are sent as the Channel Protocol specific content of inter-RBridge Channel message TRILL Data packets. Responses are sent with the same Data Label and priority as the request to which they correspond except that the response priority is limited to be not more than a configured value. This priority limit is configurable at a per RBridge level and defaults to priority 6. The Channel protocol specific data format is as follows:



V, T: Version and Type as specified in Section 3.1.

F: The Flood bit. If zero, the reply is to be unicast to the provided Nickname. If T=2, F=1 is used to flood messages for certain unsolicited update cache consistency maintenance messages from an end station Pull Directory server as discussed in Section 3.5. If T is not 2, F is ignored.

P, N: Flags used in connection with certain flooded unsolicited cache consistency maintenance messages. Ignored if T is not 2. If the P bit is a one, the solicited response message relates to cached positive response information. If the N bit is a one, the unsolicited message relates to cached negative information. See Section 3.5.

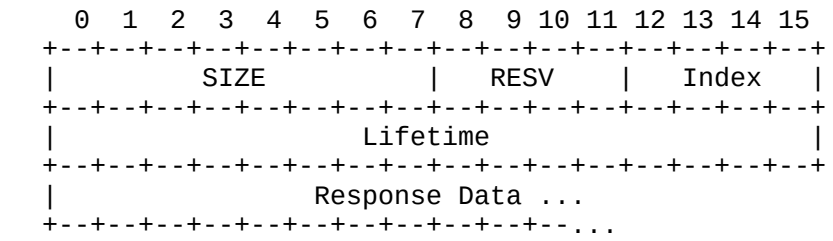
RESV: Reserved bits. MUST be sent as zero and ignored on receipt.

Count: Count is the number of responses present in the particular response message.

ERR, subERR: A two part error code. See Section 3.4.

Sequence Number: A 32-bit quantity set by the sending RBridge, returned in any responses, and used to match up responses with queries. It is opaque except that the value zero is reserved for Unsolicited Update response messages.

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



SIZE: Size of the response data in bytes starting with and including the SIZE field itself. Thus the minimum value of SIZE is 6. If SIZE is less than 6, that RESPONSE and all subsequent RESPONSEs MUST be ignored.

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

Index: The relative index of the query in the request message to which this response corresponds. The index will always be one for request messages containing a single query. The index will always be zero for unsolicited update "response" messages.

Lifetime: The length of time for which the response should be considered valid in seconds. If zero, the response can only be used for the particular query from which it resulted. The maximum time that can be expressed is just over 18.2 hours. [Perhaps this should be in units of, say, 200 milliseconds?]

Response Data: There are two types of response data.
If the ERR field is non-zero, the response data is a copy of the query data, that is, either an AFN followed by an address or a query frame.
If the ERR field is zero, the response data is the contents of an Interface Addresses APPsub-TLV (see Section 5) without the usual TRILL GENINFO TLV type and length and without the usual IA APPsub-TLV type and length before it. The maximum size of such contents is 251 bytes in the case when SIZE is 255.

Multiple response records can appear in a response message with the same index if the answer to a query 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 RBridges. 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.

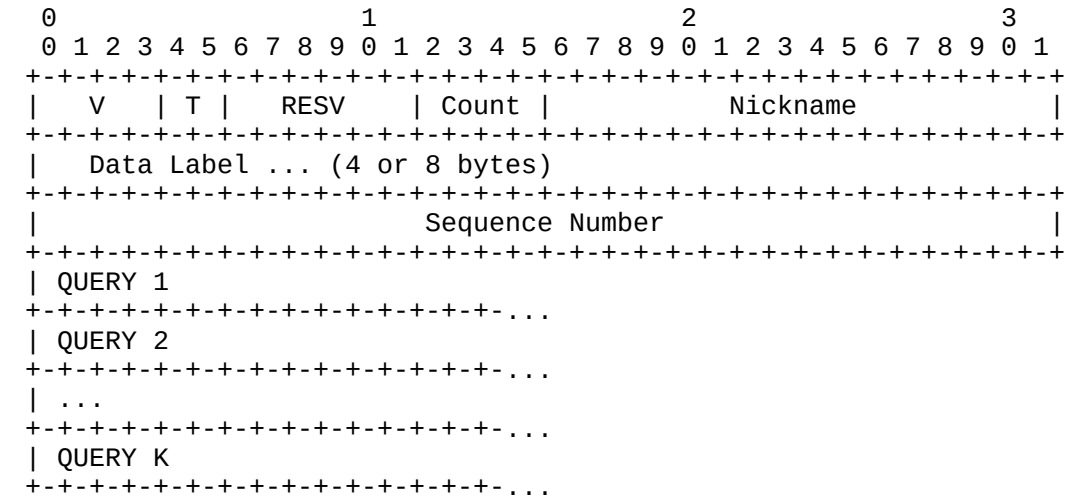
See Section 3.4 for a discussion of how errors are handled.

3.3 Pull Directory Hosted on an End Station

Optionally, a Pull Directory actually hosted on an end station MAY be supported. In that case, when the RBridge advertising itself as a Pull Directory server receives a query, 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 RBridge of the query.

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 MUST 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.

The native RBridge Channel message protocol dependent data for a Pull Directory query is formatted as follows:

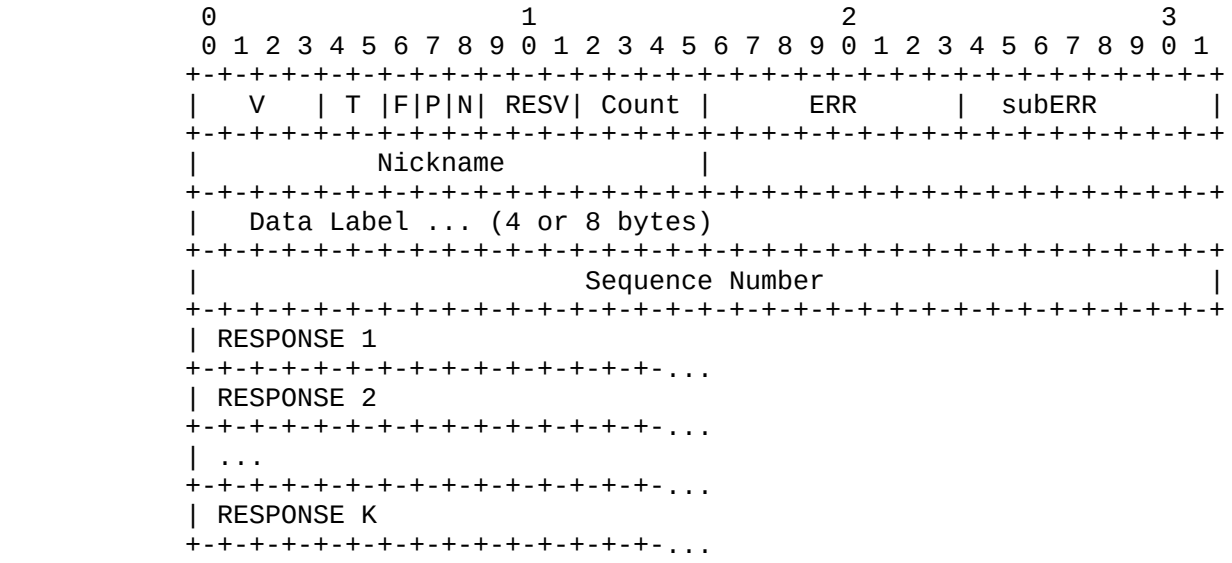


Data Label: The Data Label of the original inter-RBridge Pull Directory Channel protocol messages that was mapped to this native channel message. The format is the same as it appears right after the Inner.MacSA of the original Channel message.

Nickname: The nickname of the RBridge sending the original inter-RBridge Pull Directory query.

All other fields, including the fields within the QUERY records are as specified in Section 3.1.

The native RBridge Channel message protocol specific content for a Pull Directory response is formatted as follows:



Nickname: If F=0, the nickname of the ultimate destination RBridge. If F=1, ignored.

Data Label: The Data Label to which the response applies. The format is the same as it appears right after the Inner.MacSA in TRILL Data messages.

All other fields, including the fields within the RESPONSE records, are as specified in Section 3.2.

3.4 Pull Directory Request Errors

An error response message is indicated by a non-zero ERR field.

If there is an error that applies to an entire request 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 but expanded with a zero Lifetime and the insertion of the Index field.

If errors occur at the query level, they MUST be reported in a response message separate from the results of any successful queries.

If multiple queries in a request have different errors, they MUST be reported in separate response messages. If multiple queries in a request have the same error, this error response MAY be reported in one response message.

In an error response message, the query or queries 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.

ERR values 1 through 127 are available for encoding request message level errors. ERR values 128 through 254 are available for encoding query 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 field value
2	Request data too short
3-127	(Available for allocation by IETF Review)
128	Unknown AFN
129	Address not found
130-254	(Available for allocation by IETF REview)
255	Reserved

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

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 IETF Review)
255	Reserved

More TBD

3.5 Cache Consistency

Pull Directories MUST take action to minimize the amount of time that an RBridge will continue to use stale information from the Pull Directory.

A Pull Directory server MUST maintain one of the following, in order of increasing specificity. Retaining more specific records, such as that given in item 3 below, minimizes spontaneous response messages sent to update pull client RBridge caches. Retaining less specific records, such as that given in item 1, will generally increase the volume and overhead due to spontaneous response messages but still maintain consistency.

1. An overall record per Data Label of when the last positive response data will expire at a requester and when the last negative response will expire.

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 expected response with that data or negative response will expire at a requester.

3. For each unit of data held by the server and each address about which a negative response was sent, a list of RBridges that were sent that data as the response or sent a negative response for the address, with the expected time to expiration at each of them.

A Pull Directory server may have a limit as to how many RBridges it can maintain expiry information for 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 RBridge, the Pull Directory MUST send an unsolicited message as discussed below.

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 message is flooded (multicast) within that Data Label. And if data is added and there are outstanding cached negative responses, an all-addresses flush negative message is flooded. "All-addresses" is indicated by the Count in an unsolicited response being zero. On receiving an all-addresses flooded flush positive message from a Pull Directory server it has used, indicated by the U, F, and P bits being one, an RBridge discards all cached data responses it has for that Data Label. Similarly, on receiving an all addresses flush negative message, indicated by the U, F, and N bits being one, it discards all cached negative responses for that Data Label. A combined flush positive and negative can be flooded by having all of the U, 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 an RBridge floods address specific unsolicited update positive responses when data which is cached by a querying RBridge is changed or deleted and floods an address specific unsolicited update negative response when such information is added to. Such messages are similar to the method 1 flooded unsolicited flush messages. The U and F bits will be one and the message will be multicast. However that 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 message, if it is positive the addresses in the response records in the unsolicited response are compared to the addresses about which the recipient RBridge is holding cached positive or negative information and, if they match, the cached information is updated or the negative information replaced with the new positive information. On receiving an address specific unsolicited update negative response, the addresses in the response records in the unsolicited response are compared to the addresses about which the recipient RBridge is holding cached positive or negative information and, if they match, the any cached positive information is discarded.

If method 3 is being followed, the same sort of unsolicited update messages are sent as with method 2 except they are not normally flooded but unicast only to the specific RBridges the server believes may be holding the cached positive or negative information that may need updating. However, the Pull Directory server MAY flood the unsolicited update, for example if it determines that a sufficiently large fraction of its requesters need to be updated.

3.6 Additional Pull Details

If an RBridge notices that a Pull Directory server is no longer data reachable [RFCclear], it MUST discard all pull responses it is retaining from that server as the RBridge can no longer receive cache consistency 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 flag bit is provided as specified in Section 8.3.

4. Events That May Cause Directory Use

An RBridge 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.

Support for each of the uses 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 or a port on such a 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.)

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 RBridge is unknown, the frame must be either dropped or ingressed as a multi-destination frame which is flooded to all edge RBridges 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. It is commonly used on a link to query for the MAC address corresponding to an IPv4 address, test if an IPv4 address is in use, or 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. Gratuitous ARP
   A request used by a host to announce a new IPv4 address, new MAC address, and/or new point of network attachment. 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 RBridge 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.

3. 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 the same as General ARP queries.

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 different Ethertype 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 [ESADI] 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 which have a great number of Data Labels enabled, managing the MAC and Data Label <-> EdgeRBridge 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 RBridge Edge 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 when hosts in specific Data Labels, say VLANs 1 through 100, communicate regularly with external peers, 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 RBridges 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 RBridge 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 RBridge 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.

1. Use Push only. The RBridge subscribes to Push Directory data for Data Label X.

2. Use Pull only. When the RBridge 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 RBridge 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 modes 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.

7. Security Considerations

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

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 give IANA allocation and registry considerations.

8.1 ESADI-Parameter Data

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

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

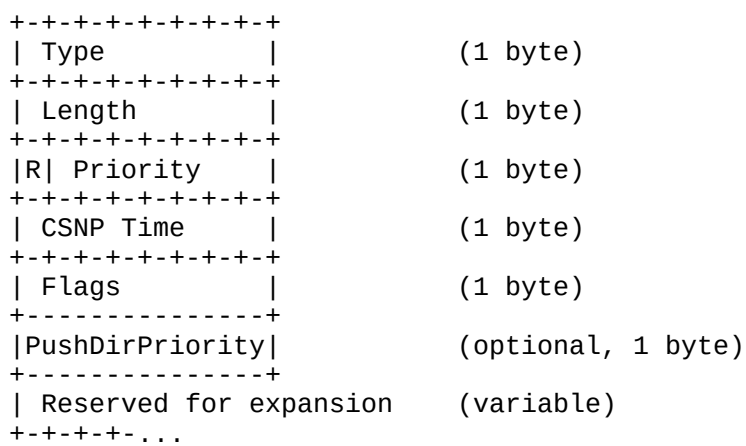
Registration Procedures: Expert Review

References: [ESADI], This document

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

The CP bit is ignored if the PD bit is zero.

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



The meanings of all the fields are as specified in [ESADI] 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 RBridge 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 table of such protocol number in the TRILL Parameters Registry referencing this document.

8.3 The Pull Directory and No Data 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 (PD) and No Data (ND) respectively. These bits are to be added to the subregistry created by [ESADI] with this document as reference.

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 an RBridge indicates connection to end stations in the VLANs or FGLs 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 for the Data Label 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" (ND) bit to one. This means that data on a distribution tree may be pruned so as not to reach the "No Data" RBridge as long as there are no RBridges interested in the Data who are beyond the "No Data" RBridge. This bit is backwards compatible as RBridges ignorant of it will simply not prune when they could, which is safe although it may cause increased link utilization.

An example of as RBridge serving as a directory that would not want multi-destination traffic in some Data Labels might be an RBridge that does not officer end station service for any of the Data Labels for which it is serving as a directory and is either (1) a Pull Directory or (2) a Push Directory for which all of the ESADI traffic can be handled by unicast [ESADI].

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.

[RFC5342bis] - Eastlake 3rd, D., "IANA Considerations and IETF Protocol Usage for IEEE 802 Parameters", BCP 141, RFC 5342, September 2008.

[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.

[ESADI] - 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.

[DirectoryFramework] - Dunbar, L., D. Eastlake, R. Perlman, I. Gashinsky, "TRILL Edge Directory Assistance Framework", draft-ietf-trill-directory-framework, in RFC Editor's queue.

[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: (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

Copyright, Disclaimer, and Additional IPR Provisions