INTERNET-DRAFT
Intended status: Proposed Standard

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

Expires: August 24, 2012

February 25, 2013

TRILL: Directory Assistance Mechanisms
<draft-dunbar-trill-scheme-for-directory-assist-04.txt>

Abstract

This document describes optional 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.

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. 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 optional specific mechanisms for directory servers to assist TRILL edge nodes.

The information held by the directories is address mapping information. Most commonly, what MAC address corresponds to an IP address within a Data Label (VLAN or FGL (Fine Grained Label [RFCfgl])) and what egress TRILL switch (RBridge) that MAC address is attached to. But it could be what IP address corresponds to a MAC address or possibly other mappings. In the data center environment, it is common for orchestration software to know and control where all the IP addresses, MAC address, and VLANs/tenants are. Thus such orchestration software is appropriate for providing the directory function or for supplying the Directory(s) with information they need.

Directory services can be offered in a Push or Pull mode. Push mode, in which a directory server pushes information to RBridges indicating interest, is specified in Section 2. Pull mode, in which an RBridge queries a server for the information it wants, is specified in Section 3. Hybrid Push/Pull modes of operation are discussed in Section 4.

The mechanisms used to keep the mappings held by different Directories synchronized is beyond the scope of this document.

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 additional acronyms and terms:

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.

RBridge: An alternative name for a TRILL switch.

TRILL switch: An alternative name for an RBridge.

1.2 Circumstances Causing Directory Use

While an RBridge can consult Directory information whenever it wants, by searching through information that has been pushed to it or requesting information from a pull directory, the following are expected to be the most common circumstances leading to directory use. All of these involve cases of ingressing a native frame.

- Ingressing an frame with an unknown unicast destination MAC. The mapping from the destination MAC and Data Label to its egress RBridge of attachment is needed to ingress the frame as unicast. If the egress RBridge is unknown, the frame must be dropped or ingressed as a multi-destination frame and flooded to all edge RBridges for its Data Label.

- Ingressing an ARP [RFC826]. ...TBD

- Ingressing a ND [RFC903]. ...TBD... Secure Neighbor Discovery messages [] 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 the ND packet rather than multicasting it.

- Ingressing a RARP [RFC4861]. ...TBD

2. Push Model Directory Assistance Mechanisms

In the Push Model, Push Directory servers push down the mapping information for the various addresses of end stations in some Data Label. A Push Directory advertises whether or not it believes it is pushing complete mapping information for a Data Label. The Push Model uses the [ESADI] protocol.

With this model, it is RECOMMENDED that complete address mapping information for a Data Label be pushed and that a participating RBridge simply drop a data packet, instead of flooding the packet, if the destination unicast MAC address is in a category being pushed and can't be found in the mapping information available. 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 the IS-IS link state database, all the Data Labels for which they are participating in [ESADI]. They are then pushed the mapping information for all such Data Labels being served by a Push Directory server.

2.2 Actions by Push Directory Servers

Push Directory servers advertise their availability to push the mapping information for a particular Data Label to ESADI participants for that Data Label by turning on a flag bit in their ESADI Parameter APPsub-TLV [ESADI] (see Section 7.1).

Each Push Directory server MUST participate in ESADI for the Data Labels for which it can push mappings and set the PD 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, as to the number of copies it believes should be pushed. Each Push Directory server also has a priority that is its 6-byte IS-IS System ID treated as an unsigned integer where larger magnitude means higher priority.

For each Data Label it can serve, each Push Directory RBridge server

orders the Push Directory servers that it can see as data reachable

[RFCclear] in the ESADI link state database for that Data Label and determines its position in that order. If a Push Directory server believes that N copies of the mappings for a Data Label should be pushed and finds that it is first in priority or, more generally, not lower than Nth in priority, it is Active. If it finds that it is N+1st or lower in priority, it is Passive.

For example, assume four Push Directory servers for Data Label X: server A with priority 123 configured to believe there should be 2 copies pushed; server B, priority 88, 1 copy; server C, priority 40, 3 copies; and server D, priority 7, 2 copies. Server A, seeing that is highest priority, is Active. Server B, seeing that it is 2nd highest priority and believing that only 1 copy should be pushed, is Passive. Server C sees that it is 3rd highest priority and believes 3 copies should be pushed, so it is Active. And server D sees it is 4th highest priority and, believing that only 2 copies should be pushed, is Passive.

If a Push Directory server is Active for Data Label X, it includes the Data Label X directory mappings it has in its ESADI-LSP for Data Label X and updates that information as the mappings it knows change. If the Push Directory server is configured to believe it has complete mapping information for Data Label X then, after it has actually transmitted all of its ESADI-LSPs for X it waits its CSNP time (see Section 6.1 of [ESADI]), and then updates its ESADI-Parameters APPsub-TLV to set the Complete Push (CP) bit to one. It then maintains the CP bit as one as long as it is Active.

If a Push Directory server is Passive for Data Label X, it removes or continues to leave out all Data Label X directory mappings it holds from its ESADI-LSP for Data Label X. However, if it was Active and was advertising the CP bit as one in its ESADI-Parameters APPsub-TLV, it first updates the CP bit to zero and sends its updated ESADI-LSP fragment zero and then waits its CSNP time before withdrawing all its directory mapping information.

2.3 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 specified in Section 5 and are flagged as being Push Directory data.

RBridges, whether or not they are a Push Directory server, MAY 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 would generally not be done as it

should 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. In that case, it SHOULD advertise the missing information unless configured not to.

Because a Push Directory server may need to advertise interest in Data Labels even though it does not want to received user data in those Data Labels, the No Data flag bit is provided as discussed in Section 7.3.

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.

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.

3. Pull Model Directory Assistance Mechanisms

In the Pull Model, an RBridge pulls mapping information from an appropriate Directory Server when needed.

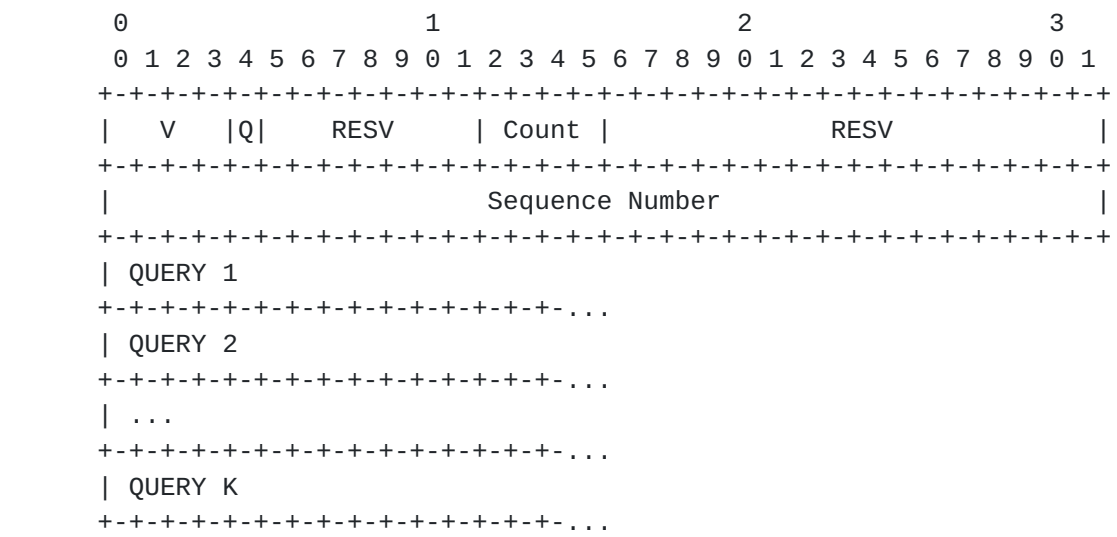
Pull Directory servers for a particular Data Label X are located by looking in the main TRILL IS-IS link state database for RBridges that advertise themselves by having the Pull Directory flag on in their Interested VLANs or Interested Labels sub-TLV [rfc6326bis] for X. If multiple RBridges indicate that they are Pull Directory Servers for a particular Data Label a pull request can be sent to any of them that is data reachable but it is RECOMMENDED that pull requests be sent to server that is least 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 7.2). Responses are returned in an RBridge Channel message using the same channel protocol number.

The requests to Pull Directory Servers are derived from normal ARP [RFC826], ND [RFC4861], RARP [RFC903] messages or data frames with unknown unicast destination MAC addresses intercepted by the RBridge when they would otherwise be ingressed. 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 or the requester is administratively prohibited from receiving the data or the like. 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 TRILL Data packet. The Data Label in the packet is the Data Label in which the address is being looked up. The priority of the channel message is a mapping of the priority frame being ingressed that caused the request with the default mapping depending, per Data Label, on the strategy (see Section 4). 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.

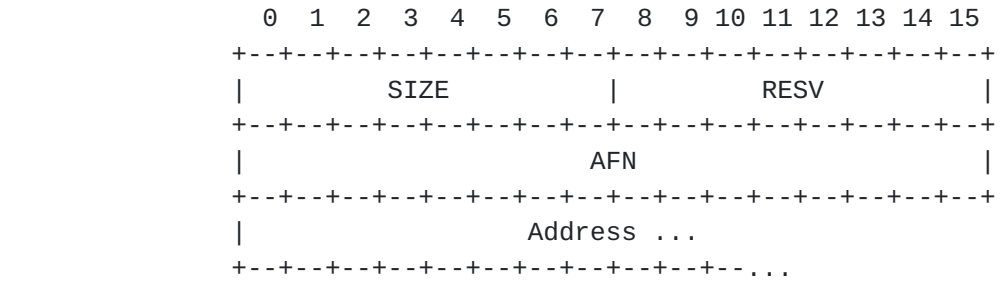
Q: Query/Response Bit. MUST be one for a query.

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

Count: Number of queries present.

Sequence Number: An opaque 32-bit quantity set by the sending RBridge, returned in any responses, and used to match up responses with queries.

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



SIZE: Size of the query data in bytes. This is the length of the Address plus 4.

RESV: A reserved byte. MUST be sent as zero and ignored on receipt.

AFN: Address Family Number of the Address.

Address: This is the address for which the query is asking for

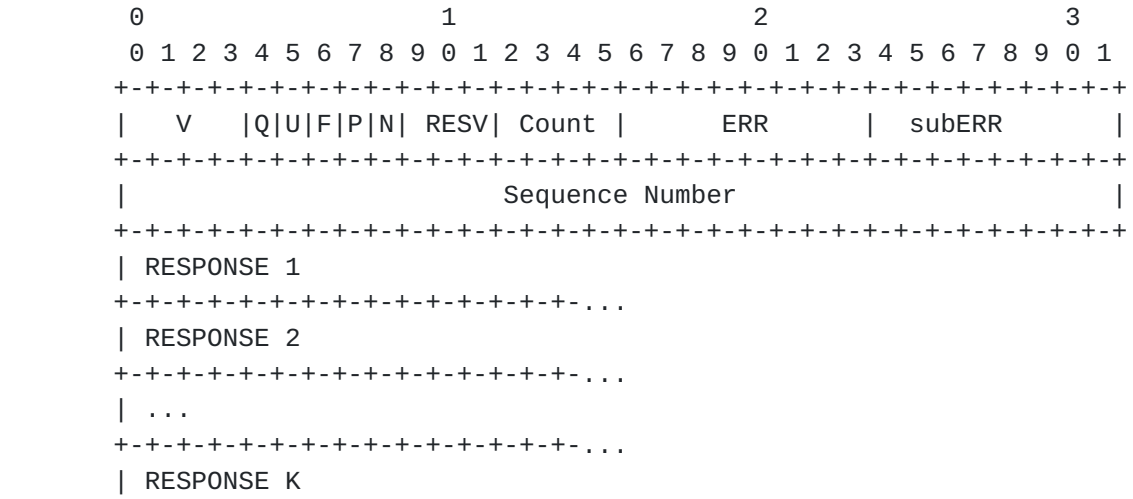
mappings. Typically it would be either (1) a MAC address, in which case the querying RBridge is interested in the RBridge by which that MAC address is reachable, or (2) an IP address, in which case the querying RBridge is interested in the corresponding MAC address and the RBridge by which that MAC address is reachable.

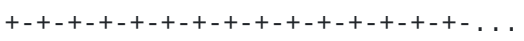
A query count of zero is explicitly allowed, for the purpose of pinging a Pull Directory server to see if it is responding to requests. It results in a response message that also has a count of zero.

If no response is received to a Pull Directory request within a configurable timeout, 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. 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: Version of the Pull Directory protocol. Version zero is specified in this document.

Q: Query/Response Bit. MUST be zero for a response.

U: Unsolicited Bit. MUST be zero for a response to a query and one for an unsolicited "response" sent to maintain cache consistency (see Section 3.5).

F: The Flood bit. If zero, the reply is to be unicast to the provided Nickname. If U=1, F=1 is used to flood messages for certain unsolicited cache consistency maintenance messages from an end station Pull Directory server as discussed in Section 3.5. If U=0, F is ignored.

P, N: Flags used in connection with certain flooded unsolicited cache consistency maintenance messages. Ignored if U is zero. 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 messages related to cached negative information. See Section 3.5.

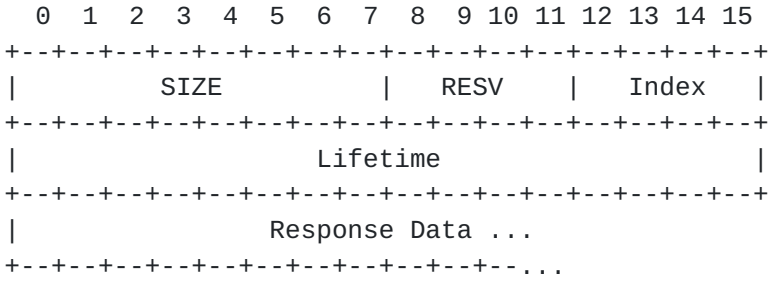
RESV: Reserved bits. MUST be sent as zero and ignored on receipt.

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

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

Sequence Number: An opaque 32-bit quantity set by the requesting RBridge and copied by the Pull Directory into all responses to the query. For an unsolicited "response", the contents are unspecified.

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



SIZE: Size of the response data in bytes plus 4.

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 "response" messages.

Lifetime: The length of time for which the response should be considered valid in seconds.

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, an AFN followed by an address. 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.

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.

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.

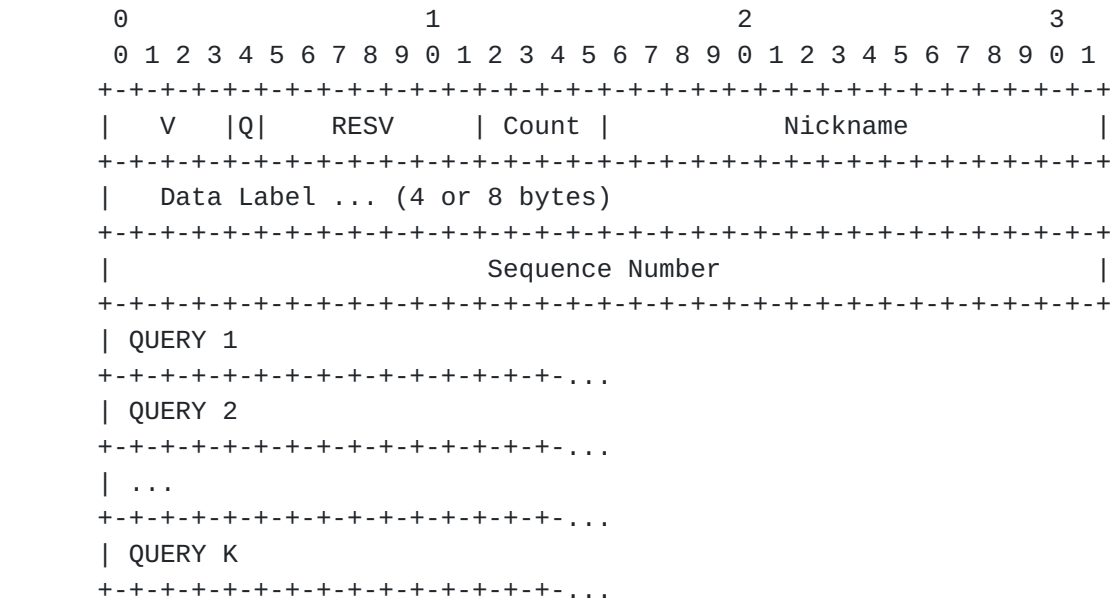
See Section 3.4 for a discussion of how errors are handled.

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 RBridge Channel Pull Directory messages use the same Channel protocol number as do the inter-RBridge Pull Directory Channel messages. The native messages MUST be sent with an Outer.VLAN tag which give 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:

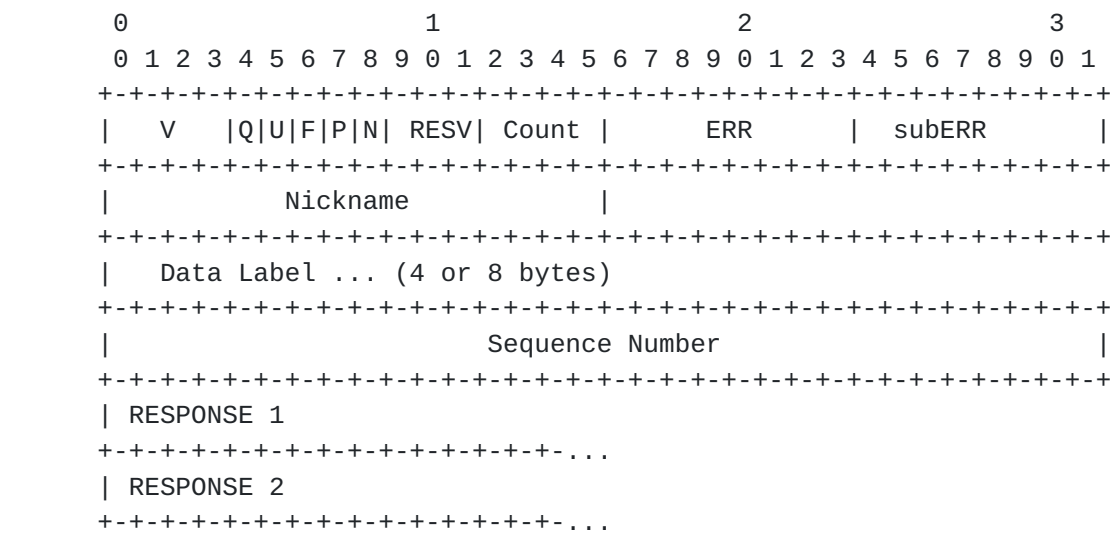


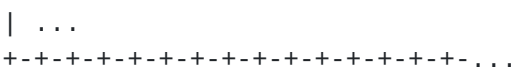
Data Label: The Data Label of the original inter-RBridge Pull Directory Channel protocol messages that was mapped to this native channel message. The format is the same as it appears right after the Inner.MacSA of the original Channel message.

Nickname: The nickname of the requesting RBridge.

All other fields are as specified in Section 3.1.

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





```
| RESPONSE K
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-...
```

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.

Nickname: The nickname of the destination RBridge or, if F=1, ignored.

All other fields 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 the 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 expanded with a zero Lifetime and the insertion of the Index field echoed back in the response records.

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 63 are available for encoding request message level errors. ERR values 64 through 255 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 V field value
2	Request data too short
3	Administratively prohibited
4-31	(Available for allocation by Standards Action)
32	Unknown AFN
33	No mapping found
34	Administratively prohibited
35-255	(Available for allocation by Standards Action)

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.

1. An overall record per Data Label of when the last returned query data will expire at a requestor and when the last query record specific negative response will expire.

2. For each unit of data (IA APPsub-TLV Address Set) held by the server and each address about which a negative response was sent, when the last expected response with that unit 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 unit as the response or sent a negative response to 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 querying 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 information in a Data Label is changed or deleted or an additional administrative Pull Directory access restriction imposed, and there are outstanding cached positive query data response(s), an all-addresses flush positive message is flooded (multicast) within that Data Label. And if data is added or an administrative restriction is removed 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 update positive unsolicited responses when data which is cached by a querying RBridge is changed or deleted or an administrative restriction is added to such data and floods an address specific update negative unsolicited responses when such information is deleted or an administrative restriction is removed from such data. 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 information and, if they match, the cached information is updated and its remaining cache life set to the minimum of its previous value in the cache and the Lifetime value in the unsolicited response. In the case of a newly imposed administrative restriction, the Lifetime in the unsolicited response is set to zero so the cached information immediately expired. On receiving an address specific unsolicited 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 negative information and, if they match, the cached negative information is discarded.

If method 3 is being followed, the same sort of messages are sent as

with method 2 except they are not flooded but unicast only to the specific RBridges the server believes may be holding the cached

positive or negative information that may need updating.

3.6 Additional Pull Details

If an RBridge notices that a Pull Directory server is no longer data reachable [RFCclear], it MUST discard all responses it is retaining from that server within one second 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 user data in those Data Labels, the No Data flag bit is provided as discussed in Section 7.3.

4. Directory Use Strategies and Push-Pull Hybrids

For some edge nodes which have great number of Data Labels enabled, managing the MAC&Label <-> RBridgeEdge 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 once a day (or once a few days) 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.

4.1 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, including ARP, ND, RARP, and unknown MAC destination addresses, 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 positive

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,

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 positive 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. Otherwise, the same as A.

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

In addition, the Pull Directory priority for an Pull Directory requests sent can be configured on a per Data Label, per ingressed frame priority basis. The default mappings are as follows:

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

network connectivity and so is avoided by default for directory

traffic.

5. The Interface Addresses APPsub-TLV

[[[This Section 5 is fairly long and complex. Should it be a separate document?]]]

This section specifies a TRILL APPsub-TLV that enables the convenient representation of sets of addresses of different types such that all of the addresses in each set designate the same end station interface (port). For example, an EUI-48 MAC (Extended Unique Identifier 48-bit, Media Access Control [RFC5342]) address, IPv4 address, and IPv6 address can be reported as all three corresponding to the same interface. This APPsub-TLV is used inside the TRILL GENINFO TLV as specified in [ESADI] and the value portion is used inside Pull Directory responses as specifies in Section 3.

Although, in some IETF protocols, address field types are represented by EtherType [RFC5342] or Hardware Type [RFC5494] only Address Family Number is used in this APPsub-TLV.

5.1 Format of the Interface Addresses APPsub-TLV

The Interface Addresses APPsub-TLV is used to indicate that a set of addresses indicate the same end-station interface and to associate that interface with the TRILL switch by which the interface is reachable. These addresses can be in different address families. For example, it can be used to declare that an end-station interface with a particular IPv4 address, IPv6 address, and EUI-48 MAC address is reachable from a particular TRILL switch.

The Template field value indicates certain well known sets of addresses or gives the number of AFNs following. When AFNs are listed, the set of AFNs provides a template for the type and order of addresses in each Address Set.

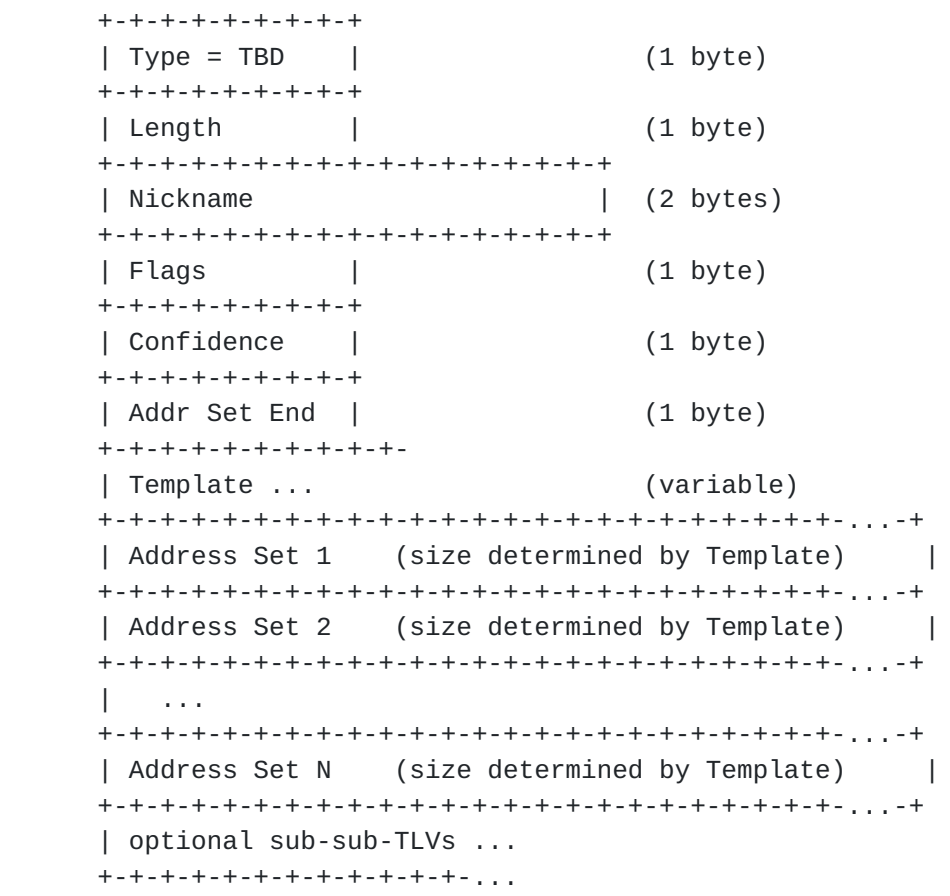


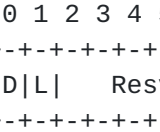
Figure 1. The Interface Addresses APPsub-TLV

- Type: Interface Addresses TRILL APPsub-TLV type, set to TBD[#2 suggested] (IA-SUBTLV).

- Length: Variable, minimum 5. If length is 4 or less, the APPsub-TLV MUST be ignored.

- Nickname: The nickname of the RBridge by which the address sets are reachable.

- Flags: A byte of flags as follows:



D: If D is one, the APPsub-TLV contains Push Directory information.

L: If L is one, the APPsub-TLV contains information learned locally be observing ingressed frames. (Both D and L can one

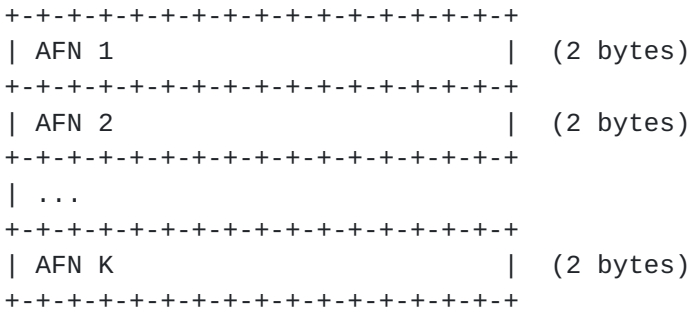
in the same APPsub-TLV.)

Resv: Additional reserved flag bits that MUST be sent as zero and ignored on receipt.

- Confidence: This 8-bit quantity indicates the confidence level in the addresses being transported [RFC6325].

- Addr Set End: The unsigned offset of the byte, within the TLV value part, of the last byte of the last Address Set. This will be the byte just before the first sub-TLV if any sub-TLVs are present. [RFC5305]

- Template: The initial byte of this field is the unsigned integer K. It K has a value from 1 to 63, it indicates that this initial byte is followed by a list of K AFNs (Address Family Numbers) in the template specifying the structure and order of each Address Set occurring later in the TLV. The minimum valid value is 1. If K is 64 to 255, it indicates that the Template for each Address Set is a specific well known Template. If the Template includes explicit AFNs, they look like the following.



- AFN: A two-byte Address Family Number. The number of AFNs present is given in first byte of the Template field if that value is less than 64. This sequence specifies the structure of the Address Sets occurring later in the TLV. For example, if Template Size is 2 and the two AFNs present are the AFNs for IPv4 and EUI-48, in that order, then each Address set present will consist of a 4-byte IPv4 address followed by a 6-byte MAC address. If any AFNs are present that are unknown to the receiving IS and the length of the corresponding address is not provided by a sub-TLV as specified below, the receiving IS will be unable to parse the Address Sets and MUST ignore the enclosing TLV.

- Address Set: Each address set consists of a sequence of addresses of the types given by the Template earlier in the TLV. No alignment, other than to a byte boundary, is guaranteed. The addresses in each Address Set are contiguous with no unused bytes between them and the Address Sets are contiguous with no unused bytes between Address Sets. The Address Sets must fit within the

TLV. If the product of the size of an Address Set and the number of Address Sets is so large that this is not true, the APPsub-TLV

is ignored.

o sub-sub-TLVs: If the Address Sets indicated by Addr Sets End do not completely fill the Length of the TLV, the remaining bytes are parsed as sub-sub-TLVs [RFC5305]. Any such sub-sub-TLVs that are not known to the receiving RBridge are ignored. Should this not be possible, for example there is only one remaining byte or an apparent sub-sub-TLV extends beyond the end of the TLV, the containing IA-APPsub-TLV is considered corrupt and is ignored. Several sub-sub-TLV types are specified in Section 5.2.

Different IA-APPsub-TLVs within the same or different EADI-LSPs or Pull Directory response from the same RBridge may have different Templates. The same AFN may occur more than once in a Template and the same address may occur in more than one address set. For example, an EUI-48 MAC address interface might have three IPv6 addresses. This could be represented by an IA-APPsub-TLV whose Template specifically provided for one EUI-48 address and three IPv6 addresses, which might be an efficient format if there were multiple interfaces with that pattern. Alternatively, a Template with one EUI-48 and one IPv6 address could be used in an IA-APPsub-TLV with three address sets each having the same EUI-48 address but different IPv6 addresses, which might be the most efficient format if only one interface had multiple IPv6 addresses and other interfaces had only one IPv6 address.

In order to be able to parse the Address Sets, a receiving RBridge must know at least the size of the address each AFN in the Template specifies; however, the presence of the Addr Set End field means that the sub-TLVs, if any, can always be located by a receiving IS. An RBridge can be assumed to know the size of IPv4 and IPv6 addresses (AFNs 1 and 2) and the size of the additional AFNs allocated by the IANA Considerations below. Should an RBridge wish to include an AFN that some receiving RBridge in the campus may not know, it SHOULD include an AFN-Size sub-sub-TLV as described below. If an IA-APPsub-TLV is received with one or more AFNs in its template for which the receiving RBridge does not know the length and for which an AFN-Size sub-sub-TLV is not present, that IA-APPsub-TLV will be ignored.

5.2 IA-APPsub-TLV sub-sub-TLVs

IA-APPsub-TLVs may have trailing sub-sub-TLVs [RFC5305] as specified below. These sub-sub-TLVs occur after the Address Sets and the amount of space available for sub-sub-TLVs is determined from the overall IA-APPsub-TLV length and the value of the Addr Set End byte.

There is no ordering restriction on sub-sub-TLVs. Unless otherwise

specified each sub-sub-TLV type can occur zero, one, or many times in

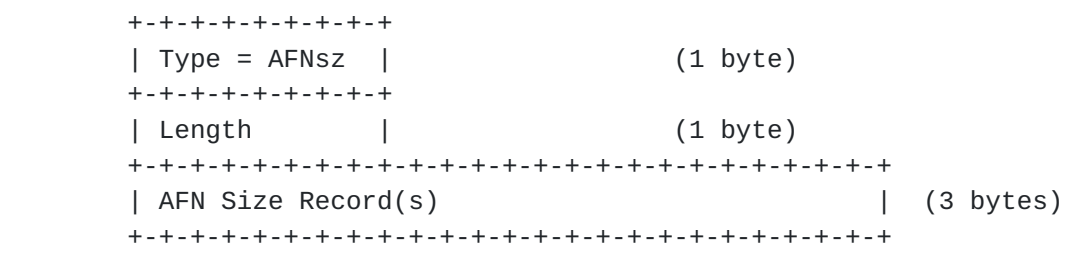
an IA-APPsub-TLV.

5.2.1 AFN Size sub-sub-TLV

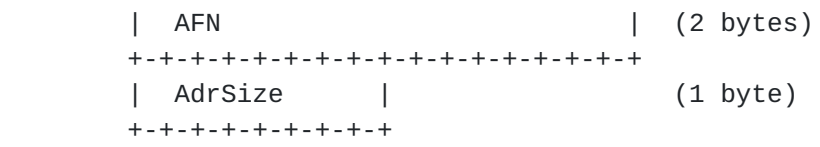
Using this sub-TLV, the originating RBridge can specify the size of an address type. This is useful under two circumstances:

1. One or more AFNs that are unknown to the receiving RBridge appears in the template. If an AFN Size sub-sub-TLV is present for each such AFN, the at least the IA-APPsub-TLV can be parses the Address Sets and make use of any address types present that it does understand.

2. If an AFN occurs in the Template that represents a variable length address, this sub-sub-TLV gives its size for all occurrences in that IA-APPsubTLV.



Where each AFN Size Record is structured as follows:



- Type: AFN-Size sub-sub-TLV type, set to 1 (AFNsz).

- Length: 3*n where n is the number of AFN Size Records present. If n is not a multiple of 3, the sub-sub-TLV MUST be ignored.

- AFN Size Record(s): Zero or more 3-byte records, each giving the size of an address type identified by an AFN,

- AFN: The AFN whose length is being specified by the AFN Size Record.

- AdrSize: The length of the address specified by the AFN field.

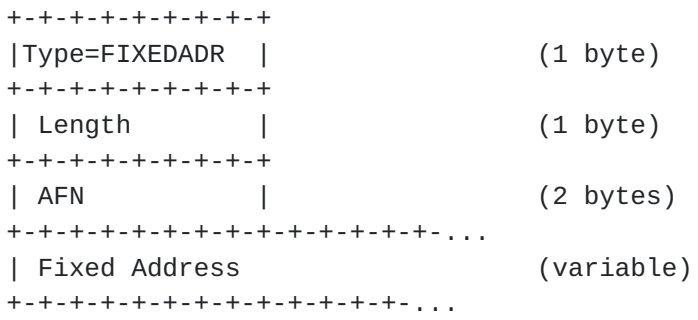
This sub-sub-TLV may occur multiple times in an enclosing IA-APPsub-

TLV.

An AFN Size sub-sub-TLV for any AFN known to the receiving RBridge (which always includes AFN 1 and 2 and the AFNs specified in xxx) is compared with the size known to the RBridge and if they differ, the IA-APPsub-TLV is ignored.

5.2.2 Fixed Address sub-sub-TLV

There may be cases where, in an Interface Addresses TLV, the same address would appear across every address set in the TLV. To avoid having a larger template and wasted space in all Address Sets, this sub-sub-TLV can be used to indicate such a fixed address



- Type: Data Label sub-sub-TLV type, set to 2 (FIXEDADR).

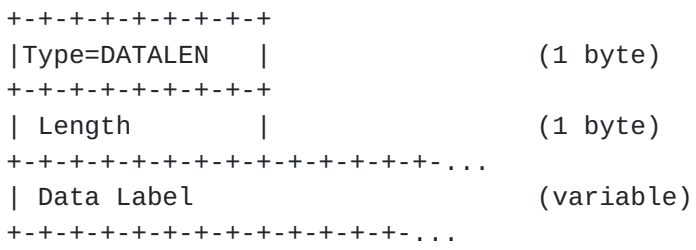
- Length: variable, minimum 3. If Length is 2 or less, the sub-sub-TLV MUST be ignored.

- AFN: Address Family Number of the Fixed Address.

- Fixed Address: The address of the type indicated by the preceding AFN field that is considered to be part of every Address Set in the IA-APPsub-TLV.

5.2.3 Data Label sub-sub-TLV

When used with Push or Pull Directories, the Data Label is indicated by the Data Label of the ESADI instance (Push) or RBridge Channel message (Pull) in which the IA APPsub-TLV appears and any occurrence of this sub-sub-TLV is ignored. However, the IA APPsub-TLV might be used in other contexts where this sub-sub-TLV indicates the Data Label of the Address Sets and multiple occurrences of this sub-sub-TLV indicate that the Address Sets exist in all of the Data Labels.



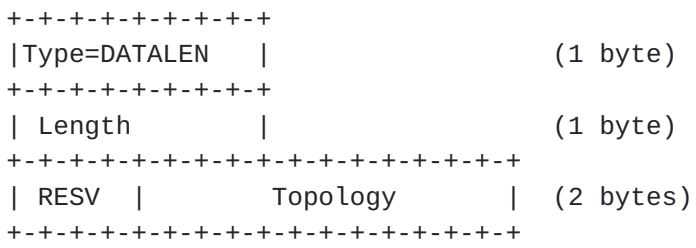
o Type: Data Label sub-TLV type, set to 3 (DATALEN).

o Length: 2 or 3

o Data Label: If length is 2, the bottom 12 bits of the Data Label are a VLAN ID and the top 4 bits are reserved (MUST be sent as zero and ignored on receipt). If the length is 3, the three Data Label bytes contain an FGL [RFCfgl].

5.2.4 Topology sub-sub-TLV

The presence of this sub-sub-TLV indicates that the Address Sets are in the topology give. If it occurs multiple times, then the Address Sets are in all of the topologies listed.



o Type: Data Label sub-TLV type, set to 3 (DATALEN).

o Length: 2.

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

o Topology: The 12-bit topology number.

6. 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] and [RFC5310].

Pull Directory queries and responses are transmitted as RBridge-to-RBridge or native RBridge Channel messages. Such messages can be secured by TBD

For general TRILL security considerations, see [RFC6325].

7. IANA Considerations

This section give IANA allocation and registry considerations.

7.1 ESADI-Parameter Bits

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

Sub-Registry: ESADI-Parameter APPsub-TLV Bits

Registration Procedures: IETF 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	

7.2 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.

7.3 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 3 and 4) and the Interested Labels field of the Interested Labels sub-TLV (suggested bits 5 and 6) [rfc6326bis] to indicate Pull Directory server (PD) and No Data (ND) respectively. These bits are to be added to the subregistry set up in [ESADI].

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 although multicast traffic

might be pruned. But, with Push and Pull Directories, advertising that you are a directory server requires using these sub-TLVs as part of advertising that you are a directory server. If such a directory server does not wish to received multi-destination user data 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 no prune when it could, which is safe but may cause increased link utilization.

7.4 Additional AFN Number Allocation

IANA is requested to allocate four new AFN numbers as follows:

Number	Description	References
TBD(26)	EUI-48	RFC 5342, this document
TBD(27)	OUI	RFC 5342, this document
TBD(28)	MAC/24	This document.
TBD(29)	IPv6/64	This document.

The OUI AFN is provided so that MAC addresses can be abbreviated if they have the same upper 24 bits. In particular, if there is an OUI provided as a Fixed Address sub-sub-TLV (see Section 5.2.2) then, whenever a MAC/24 address appears within an Address Set (as indicated by the Template), the OUI is used as the first 24 bits of the actual MAC address for the Address Set.

MAC/24 is a 24-bit suffixes intended to be pre-fixed by an OUI as in the previous paragraph. In absence of an OUI specified as a Fixed Address in the same APPsub-TLV, the Address Set cannot be used.

IPv6/64 is an 8-byte quantity that is the first 64 bits of an IPv6 address. If present, there will normally be an EUI-64 address in the address set to provide the lower 64 bits of the IPv6 address. For this purpose, an EUI-48 is expanded to 64 bits as described in [RFC5342].

7.5 IA APPsub-TLV Sub-Sub-TLVs SubRegistry

IANA is requested to establish a new subregistry for sub-sub-TLVs of the Interface Addresses APPsub-TLV with initial contents as shown

below.

Name: Interface Addresses APPsub-TLV Sub-Sub-TLVs

Procedure: IETF Review

Reference: This document

Type	Description	Reference
0	Reserved	
1	AFN Size	This document
2	Fixed Address	This document
3	Data Label	This document
4	Topology	This document
5-254	Available	This document
255	Reserved	

8. Acknowledgments

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

9. References

Normative and Informational References are given below.

9.1 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

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

[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.

[RFC5305] - Li, T. and H. Smit, "IS-IS Extensions for Traffic Engineering", RFC 5305, October 2008.

[RFC5342] - Eastlake 3rd, D., "IANA Considerations and IETF Protocol Usage for IEEE 802 Parameters", BCP 141, RFC 5342, September 2008.

[RFC5494] - Arkko, J. and C. Pignataro, "IANA Allocation Guidelines for the Address Resolution Protocol (ARP)", RFC 5494, April 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.

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

rfc6326bis-00.txt, 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-05.txt, work in progress.

[ESADI] - Zhai, H., F. Hu, R. Perlman, D. Eastlake, J. Hudson, "TRILL (Transparent Interconnection of Lots of Links): The ESADI (End Station Address Distribution Information) Protocol", draft-ietf-trill-esadi-02.txt, work in progress.

9.2 Informational References

[RFC5342] - Eastlake 3rd, D., "IANA Considerations and IETF Protocol Usage for IEEE 802 Parameters", BCP 141, RFC 5342, September 2008

[DirectoryFramework] - Dunbar, L., D. Eastlkae, R. Perlman, I. Gashinsky, "TRILL Edge Directory Assistance Framework", draft-ietf-trill-directory-framework-03.txt, 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