

Internet Engineering Task Force (IETF)  
Request for Comments: 7961  
Category: Standards Track  
ISSN: 2070-1721

D. Eastlake 3rd  
Y. Li  
Huawei  
August 2016

## Transparent Interconnection of Lots of Links (TRILL): Interface Addresses APPsub-TLV

### Abstract

This document specifies a TRILL (Transparent Interconnection of Lots of Links) IS-IS application sub-TLV that enables the reporting by a TRILL switch of sets of addresses. Each set of addresses reports all of the addresses that designate the same interface (port) and also reports the TRILL switch by which that interface is reachable. For example, a 48-bit MAC (Media Access Control) address, IPv4 address, and IPv6 address can be reported as all corresponding to the same interface reachable by a particular TRILL switch. Such information could be used in some cases to synthesize responses to, or bypass the need for, the Address Resolution Protocol (ARP), the IPv6 Neighbor Discovery (ND) protocol, or the flooding of unknown MAC addresses.

### Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at <http://www.rfc-editor.org/info/rfc7961>.

## Copyright Notice

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction .....	3
1.1. Conventions Used in This Document .....	3
2. Format of the Interface Addresses APPsub-TLV .....	4
3. IA APPsub-TLV Sub-sub-TLVs .....	9
3.1. AFN Size Sub-sub-TLV .....	10
3.2. Fixed Address Sub-sub-TLV .....	11
3.3. Data Label Sub-sub-TLV .....	12
3.4. Topology Sub-sub-TLV .....	12
4. Security Considerations .....	13
5. IANA Considerations .....	14
5.1. Allocation of AFN Values .....	14
5.2. IA APPsub-TLV Sub-sub-TLVs Sub-registry .....	15
5.3. IA APPsub-TLV Number .....	16
6. Additional AFN Information .....	16
7. Processing Address Sets .....	16
8. References .....	18
8.1. Normative References .....	18
8.2. Informative References .....	20
Appendix A. Examples .....	21
A.1. Simple Example .....	21
A.2. Complex Example .....	22
Acknowledgments .....	24
Authors' Addresses .....	24

## 1. Introduction

This document specifies a TRILL (Transparent Interconnection of Lots of Links) [RFC6325] IS-IS application sub-TLV (APPsub-TLV) [RFC6823] that enables the convenient representation of sets of addresses where all of the addresses in each set designate the same interface (port). For example, a 48-bit MAC (Media Access Control) [RFC7042] address, IPv4 address, and IPv6 address can be reported as all three designating the same interface. In addition, a Data Label (VLAN or Fine-Grained Label (FGL) [RFC7172]) is specified for the interface, along with the TRILL switch and, optionally, the TRILL switch port from which the interface is reachable. Such information could be used in some cases to synthesize responses to, or bypass the need for, the Address Resolution Protocol (ARP) [RFC826], the IPv6 Neighbor Discovery (ND) [RFC4861] protocol, the Reverse Address Resolution Protocol (RARP) [RFC903], or the flooding of unknown destination MAC addresses [ARPND]. If the information reported is complete, it can also be used to detect and discard packets with forged source addresses.

This APPsub-TLV appears inside the TRILL GENINFO TLV specified in the End Station Address Distribution Information (ESADI) RFC [RFC7357] but may also occur in other application contexts. The "directory assistance" TRILL Edge services [DirectoryScheme] are expected to make use of this APPsub-TLV.

Although in some IETF protocols address field types are represented by an Ethertype [RFC7042] or a hardware address type [RFC5494], only the Address Family Number (AFN) is used in this APPsub-TLV to represent the address field type.

### 1.1. Conventions Used in This Document

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 [RFC2119]. Capitalized IANA-related terms such as "Expert Review" are to be interpreted as described in [RFC5226].

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

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

APPsub-TLV: Application sub-TLV [RFC6823]

Data Label: VLAN or FGL

**FGL:** Fine-Grained Label [RFC7172]

**IA:** Interface Address(es)

**MAC:** Media Access Control

**Nickname:** A 16-bit TRILL switch identifier, as specified in Section 3.7 of [RFC6325] and as updated by Section 4 of [RFC7780]

**RBridge:** An alternative name for a TRILL switch

**TRILL switch:** A device that implements the TRILL protocol

## 2. Format of the Interface Addresses APPsub-TLV

The Interface Addresses (IA) APPsub-TLV is used to advertise a set of addresses indicating the same interface (port) within a Data Label (VLAN or FGL). It also associates that interface with the TRILL switch and, optionally, the TRILL switch port by which the interface is reachable. These addresses can be in different address families. For example, the IA APPsub-TLV can be used to declare that a particular interface with specified IPv4, IPv6, and 48-bit MAC addresses in some particular Data Label is reachable from a particular TRILL switch. While those three types of addresses are likely to be the only types of interest, any address type for which an AFN has been assigned by IANA can be represented.

The Template field in a particular IA APPsub-TLV indicates the format of each Address Set it carries. Certain well-known sets of addresses are represented by special values. Other sets of addresses are specified by a list of AFNs. The Template format that uses a list of AFNs provides an explicit pattern for the type and order of addresses in each Address Set in the IA APPsub-TLV that includes that Template.

A device or application making use of IA APPsub-TLV data is not required to make use of all IA data. For example, a device or application that was only interested in MAC and IPv6 addresses could ignore any IPv4 or other types of address information that was present.

Figure 1 shows an IA APPsub-TLV as it would appear inside an IS-IS Flooding Scope Link State PDU (FS-LSP) using an extended flooding scope [RFC7356] TLV -- for example, in ESADI [RFC7357]. Within an IS-IS FS-LSP using traditional [ISO-10589] TLVs, the Type and Length would be 1-byte unsigned integers equal to or less than 255, but with an extended TLV, the Type and Length are 2-byte unsigned integers.

```

+-----+
| Type = (10) | (2 bytes)
+-----+
| Length      | (2 bytes)
+-----+
| Addr Sets End | (2 bytes)
+-----+
| Nickname    | (2 bytes)
+-----+
| Flags       | (1 byte)
+-----+
| Confidence  | (1 byte)
+-----+
| Template ... | (variable)
+-----+
| Address Set 1 (size determined by Template) ...+
+-----+
| Address Set 2 (size determined by Template) ...+
+-----+
| ...
+-----+
| Address Set N (size determined by Template) ...+
+-----+
| optional sub-sub-TLVs ...
+-----+

```

Figure 1: Interface Addresses APPsub-TLV

- o Type: Interface Addresses TRILL APPsub-TLV type; set to 10 (IA-SUBTLV).
- o Length: Variable; minimum 7. If Length is 6 or less or if the APPsub-TLV extends beyond the size of an encompassing TRILL GENINFO TLV or other context, the APPsub-TLV MUST be ignored. For manageability, a counter reflecting the receipt of such malformed IA APPsub-TLVs should be maintained.

- o **Addr Sets End:** The unsigned integer byte number, within the IA APPsub-TLV value part, of the last byte of the last Address Set, where the first byte is numbered 1. This will be the number of the byte just before the first sub-sub-TLV if any sub-sub-TLVs are present (see Section 3). The processing is as follows:
  - If this field is greater than Length or points to before the end of the Template, the IA APPsub-TLV is corrupt and **MUST** be discarded.
  - If this field is equal to Length, there are no sub-sub-TLVs.
  - If this field is less than Length, sub-sub-TLVs are parsed as specified in Section 3.

**Note:** This field is always 2 bytes in size.

- o **Nickname:** The nickname (see Section 1.1) of the TRILL switch by which the Address Sets are reachable. If 0, the Address Sets are reachable from the TRILL switch originating the message containing the APPsub-TLV (for example, an ESADI [RFC7357] message).
- o **Flags:** A byte of flags, as follows:

```

  0 1 2 3 4 5 6 7
+---+---+---+---+
|D|L|   RESV   |
+---+---+---+---+

```

**D: Directory flag:** If D is 1, the APPsub-TLV contains directory information [RFC7067].

**L: Local flag:** If L is 1, the APPsub-TLV contains information learned locally by observing ingressed frames [RFC6325]. (Both D and L can be set to 1 in the same IA APPsub-TLV if a TRILL switch had learned an address locally and also advertised it as a directory.)

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

- o **Confidence:** This 8-bit unsigned quantity in the range 0 to 254 indicates the confidence level in the addresses being transported (see Section 4.8.2 of [RFC6325]). A value of 255 is treated as if it was 254.

- o **Template:** The initial byte of this field is the unsigned integer K. If K has a value from 1 to 31, it indicates that this initial byte is followed by a list of K AFNs that specify the exact structure and order of each Address Set occurring later in the APPsub-TLV. K can be 1, which is the minimum valid value. If K is 0, the IA APPsub-TLV is ignored. If K is 32 to 254, the length of the Template field is 1 byte, and its value is intended to correspond to a particular ordered set of AFNs, some of which are specified below. The value of 255 for K is reserved for future definition and causes the IA APPsub-TLV to be ignored.

If the Template uses explicit AFNs, it looks like the following, with the number of AFNs, up to 31, equal to K.

```

+---+---+---+---+---+---+
|  K  |                                     (1 byte)
+---+---+---+---+---+---+
|  AFN 1  |                                     (2 bytes)
+---+---+---+---+---+---+
|  AFN 2  |                                     (2 bytes)
+---+---+---+---+---+---+
|  ...  |
+---+---+---+---+---+---+
|  AFN K  |                                     (2 bytes)
+---+---+---+---+---+---+

```

For K in the range 32 to 39, values indicate a specific sequence, as specified below. The values of K from 40 to 254 are reserved for future specification. If the value of K is not understood by a receiver of the IA-APPsub-TLV, any Address Sets present are ignored.

K	Addresses in order of occurrence
---	-----
32	48-bit MAC
33	48-bit MAC, IPv4
34	48-bit MAC, IPv6
35	48-bit MAC, IPv4, IPv6
36	48-bit MAC, RBridge port
37	48-bit MAC, IPv4, RBridge port
38	48-bit MAC, IPv6, RBridge port
39	48-bit MAC, IPv4, IPv6, RBridge port

For ease of decoding, note that for values of K between 32 and 39 inclusive, the 0x01 bit indicates that an IPv4 address is present, the 0x02 bit indicates that an IPv6 address is present, and the 0x04 bit indicates that an RBridge Port ID is present.

- o **AFN:** A 2-byte Address Family Number. The number of AFNs present is given by K, except that there are no AFNs if K is greater than 31. The AFN sequence specifies the structure of the Address Sets occurring later in the TLV. For example, if the Template size is 2 and the two AFNs present are the AFNs for a 48-bit MAC and an IPv4 address, in that order, then each Address Set present will consist of a 6-byte MAC address followed by a 4-byte IPv4 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-sub-TLV as specified below, the receiving IS will be unable to parse the Address Sets and MUST ignore the IA APPsub-TLV.
- o **Address Set:** Each Address Set in the APPsub-TLV consists of exactly the same sequence of addresses and types as specified by the Template earlier in the APPsub-TLV. No alignment, other than to a byte boundary, is provided. 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 successive Address Sets. The Address Sets must fit within the TLV. See Section 7 on interpreting certain Address Sets.
- o **sub-sub-TLVs:** If the Address Sets indicated by Addr Sets End do not completely fill the length of the APPsub-TLV (as indicated by the Length field), then per Section 4 of [RFC5305] the remaining bytes are parsed as sub-sub-TLVs. Any such sub-sub-TLVs that are not known to the receiving TRILL switch are ignored. Should this parsing 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 3.)

Different IA APPsub-TLVs within the same or different LSPs or other data structures may have different Templates. The same AFN may occur more than once in a Template, and the same address may occur in different Address Sets. For example, a 48-bit MAC address interface might have three different IPv6 addresses. This could be represented by an IA APPsub-TLV whose Template specifically provided for one EUI-48 address and three IPv6 addresses; this might be an efficient format if there were multiple interfaces with that pattern. Alternatively, a Template with one 48-bit MAC and one IPv6 address could be used in an IA APPsub-TLV with three Address Sets each having the same MAC address but different IPv6 addresses; this 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 TRILL switch must know at least the size of the address for each AFN or address type the Template specifies; however, the presence of the Addr Sets End field means that the sub-sub-TLVs, if any, can always be located by a receiver. A TRILL switch can be assumed to know the size of the AFNs mentioned in Section 5. Should a TRILL switch wish to include an AFN that some receiving TRILL switch in the campus may not know, it SHOULD include an AFN Size sub-sub-TLV as described in Section 3.1. If an IA APPsub-TLV is received with one or more AFNs in its Template for which the receiving TRILL switch does not know the length and for which an AFN Size sub-sub-TLV is not present, that IA APPsub-TLV MUST be ignored.

For manageability, a counter of ill-formed IA APPsub-TLVs received and ignored due to unknown K, unknown AFN, and the like (as described above) should be maintained.

### 3. IA APPsub-TLV Sub-sub-TLVs

IA APPsub-TLVs can have sub-sub-TLVs (sub-TLVs of sub-TLVs [RFC5305]) at the end, as specified below. These sub-sub-TLVs occur after the Address Sets. The amount of space available for sub-sub-TLVs is determined from the overall IA APPsub-TLV length and the value of the Addr Sets 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. Any sub-sub-TLVs for which the Type is unknown are ignored. For manageability, a counter of sub-sub-TLVs received and ignored due to an unknown Type or other reasons, as described below, should be maintained.

The data structures of the sub-sub-TLVs shown below, with 2-byte Types and Lengths, assume that the enclosing IA APPsub-TLV is in an extended LSP TLV [RFC7356] or some non-LSP context. If they were used in an IA APPsub-TLV in a non-extended LSP [ISO-10589], then only 1-byte Types and Lengths could be used. As a result, any sub-sub-TLV types greater than 255 could not be used, and Length would be limited to 255.

### 3.1. AFN Size Sub-sub-TLV

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

1. One or more AFNs that are unknown to the receiving TRILL switch appear in the Template. If an AFN Size sub-sub-TLV is present for each such AFN, then at least the IA APPsub-TLV can be parsed, and possibly other addresses in each Address Set can still be used.
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 APPsub-TLV.

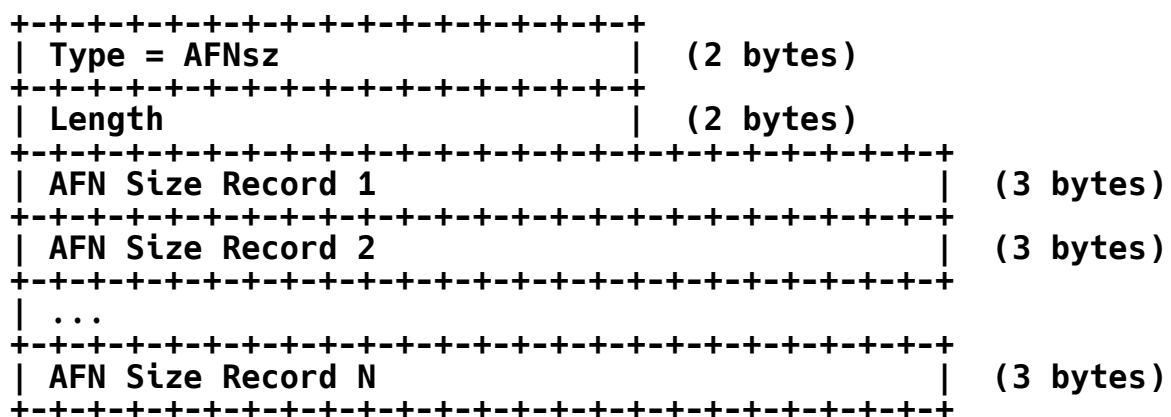
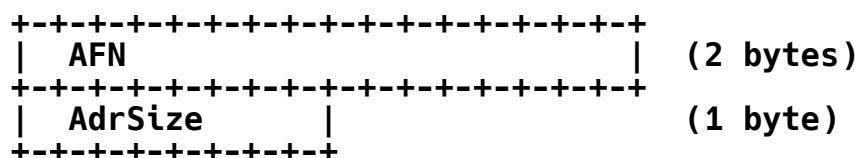


Figure 2: AFN Size Sub-sub-TLV

Where each AFN Size Record is structured as follows:



- o Type: AFN Size sub-sub-TLV type; set to 1 (AFNsz).
- o Length:  $3 \times N$ , where  $N$  is the number of AFN Size Records present. If Length is not a multiple of 3, the sub-sub-TLV MUST be ignored.
- o AFN Size Record(s): Zero or more 3-byte records, each giving the size of an address type identified by an AFN.

- o AFN: The AFN whose length is being specified by the AFN Size Record.
- o AdrSize: The length, in bytes, of addresses specified by the AFN field as an unsigned integer.

An AFN Size sub-sub-TLV for any AFN known to the receiving TRILL switch is compared with the size known to the TRILL switch. If they differ, the IA APPsub-TLV is assumed to be corrupt and **MUST** be ignored.

### 3.2. Fixed Address Sub-sub-TLV

There may be cases where, in a particular IA APPsub-TLV, the same address would appear in every Address Set across the IA APPsub-TLV. To avoid wasted space, this sub-sub-TLV can be used to indicate such a fixed address. The address or addresses incorporated into the sets by this sub-sub-TLV are **NOT** mentioned in the IA APPsub-TLV Template.

```

+---+---+---+---+---+---+---+---+---+---+---+---+
| Type = FIXEDADR                                     | (2 bytes)
+---+---+---+---+---+---+---+---+---+---+---+---+
| Length                                              | (2 bytes)
+---+---+---+---+---+---+---+---+---+---+---+---+
| AFN                                                 | (2 bytes)
+---+---+---+---+---+---+---+---+---+---+---+---+
| Fixed Address                                     (variable)
+---+---+---+---+---+---+---+---+---+---+---+---+

```

Figure 3: Fixed Address Sub-sub-TLV

- o Type: Data Label sub-sub-TLV type; set to 2 (FIXEDADR).
- o Length: Variable; minimum 2. If Length is 0 or 1, the sub-sub-TLV **MUST** be ignored.
- o AFN: Address Family Number of the Fixed Address.
- o 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.

The Length field implies a size for the Fixed Address. If that size differs from the size of the address type for the given AFN as known by the receiving TRILL switch, the Fixed Address sub-sub-TLV is considered corrupt and **MUST** be ignored.

### 3.3. Data Label Sub-sub-TLV

This sub-sub-TLV indicates the Data Label within which the interfaces listed in the IA APPsub-TLV are reachable. It is useful if the IA APPsub-TLV occurs outside of the context of a message specifying the Data Label or if it is desired and permitted to override that specification. Multiple occurrences of this sub-sub-TLV indicate that the interfaces are reachable in all of the Data Labels given.

```

+---+---+---+---+---+---+---+---+---+---+
|Type = DATALEN                               | (2 bytes)
+---+---+---+---+---+---+---+---+---+---+
| Length                                       | (2 bytes)
+---+---+---+---+---+---+---+---+---+---+
| Data Label                                  | (variable)
+---+---+---+---+---+---+---+---+---+...

```

Figure 4: Data Label Sub-sub-TLV

- o Type: Data Label sub-TLV type; set to 3 (DATALEN).
- o Length: 2 or 3. If Length is some other value, the sub-sub-TLV MUST be ignored.
- 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 Length is 3, the three Data Label bytes contain an FGL [RFC7172].

### 3.4. Topology Sub-sub-TLV

The presence of this sub-sub-TLV indicates that the interfaces given in the IA APPsub-TLV are reachable in the topology given. It is useful if the IA APPsub-TLV occurs outside of the context of a message indicating the topology or if it is desired and permitted to override that specification. If it occurs multiple times, then the Address Sets are in all of the topologies given.

```

+---+---+---+---+---+---+---+---+---+---+
|Type = TOPOLOGY                               | (2 bytes)
+---+---+---+---+---+---+---+---+---+---+
| Length                                       | (2 bytes)
+---+---+---+---+---+---+---+---+---+---+
| RESV | Topology                             | (2 bytes)
+---+---+---+---+---+---+---+---+---+---+

```

Figure 5: Topology Sub-sub-TLV

- o Type: Topology sub-TLV type; set to 4 (TOPOLOGY).
- o Length: 2. If Length is some other value, the sub-sub-TLV MUST be ignored.
- o RESV: 4 reserved bits. MUST be sent as zero and ignored on receipt.
- o Topology: The 12-bit topology number [RFC5120].

#### 4. Security Considerations

The integrity of address mapping and reachability information as well as the correctness of Data Labels (VLANs or FGLs [RFC7172]) are very important. Forged, altered, or incorrect address mapping or data labeling can lead to delivery of packets to the incorrect party, violating security policy. However, this document merely describes a data format and does not provide any explicit mechanisms for securing that information, other than a few simple consistency checks that might detect some corrupted data. Security on the wire, or in storage, for this data is to be provided by the transport or storage used. For example, when transported with ESADI [RFC7357] or RBridge Channel [RFC7178], ESADI security or Channel Tunnel [ChannelTunnel] security mechanisms can be used, respectively.

The address mapping and reachability information, if known to be complete and correct, can be used to detect some cases of forged packet source addresses [RFC7067]. In particular, if native traffic from an end station is received by a TRILL switch that would otherwise accept it but authoritative data indicates that the source address should not be reachable from the receiving TRILL switch, that traffic should be discarded. The data format specified in this document may optionally include a TRILL switch Port ID number so that this forged address filtering can be optionally applied with port granularity. For manageability, a counter of frames so discarded should be maintained.

See [RFC6325] for general TRILL security considerations.

## 5. IANA Considerations

The following subsections specify IANA allocations.

### 5.1. Allocation of AFN Values

IANA has allocated values in the "Address Family Numbers" registry that may be useful for IA APPsub-TLVs. The values are as follows:

Hex	Decimal	Description	References
-----	-----	-----	-----
0001	1	IPv4	
0002	2	IPv6	
4005	16389	48-bit MAC	Section 2.1 of [RFC7042]
4006	16390	64-bit MAC	Section 2.2 of [RFC7042]
4007	16391	OUI	Section 6 of RFC 7961
4008	16392	MAC/24	Section 6 of RFC 7961
4009	16393	MAC/40	Section 6 of RFC 7961
400A	16394	IPv6/64	Section 6 of RFC 7961
400B	16395	RBridge Port ID	Section 6 of RFC 7961

Other AFNs can be found at <<http://www.iana.org/assignments/address-family-numbers>>.

See Section 7 on interpreting Address Sets.

## 5.2. IA APPsub-TLV Sub-sub-TLVs Sub-registry

IANA has established a new sub-registry of the "Transparent Interconnection of Lots of Links (TRILL) Parameters" registry for sub-sub-TLVs of the Interface Addresses APPsub-TLV, with the following initial contents:

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

Procedure: Expert Review

Note: Types greater than 255 are not usable in some contexts.

Reference: RFC 7961

Type	Description	Reference
-----	-----	-----
0	Reserved	RFC 7961
1	AFN Size	RFC 7961
2	Fixed Address	RFC 7961
3	Data Label	RFC 7961
4	Topology	RFC 7961
5-254	Unassigned	
255	Reserved	RFC 7961
256-65534	Unassigned	
65535	Reserved	RFC 7961

Expert Guidance: A designated expert for this registry should decide whether to permit the assignment of a type based on clear documentation of the proposed type as provided by the requester, such as a complete Internet-Draft. New types should not duplicate existing types. Requests should indicate whether a type less than 255 is desired; such types can be used in contexts where only 1 byte of a type (and usually only 1 byte of the length) is permitted. Types greater than 255 can only be used where 2-byte types are allowed, such as in Extended Level 1 Flooding Scope (E-L1FS) or Extended Level 1 Circuit Scope (E-L1CS) extended FS-LSPs [RFC7356]; in those contexts, lengths up to 65535 bytes can also be expressed, although they may not be usable if the resulting TLV would not fit into a larger context restricted by an MTU setting or the like. Values within the region below 255 and the region above 255 should be allocated sequentially, unless there is an extraordinary reason for a special value.

### 5.3. IA APPsub-TLV Number

IANA has allocated type 10 as the IA APPsub-TLV in the "TRILL APPsub-TLV Types under IS-IS TLV 251 Application Identifier 1" registry from the range under 256. In the registry, the name is "IA" and the reference is this document.

## 6. Additional AFN Information

This section provides additional information concerning AFNs that were allocated in connection with this document. These AFNs are not restricted to use in the IA APPsub-TLV and may be used in other protocols where they would be appropriate.

**OUI:** A 3-byte (24-bit) Organizationally Unique Identifier used as the initial 3 bytes of a MAC address. See Sections 2.1 and 2.2 of [RFC7042], and Section 7 below.

**MAC/24:** A 3-byte (24-bit) quantity used as the final 3 bytes of a 48-bit MAC address. See Section 2.1 of [RFC7042] and Section 7 below.

**MAC/40:** A 5-byte (40-bit) quantity used as the final 5 bytes of a 64-bit MAC address. See Section 2.2 of [RFC7042] and Section 7 below.

**IPv6/64:** An 8-byte (64-bit) quantity used as the initial 8 bytes of an IPv6 address. See Section 7 below.

**RBridge Port ID:** A 16-bit quantity that uniquely identifies a port on a TRILL switch (RBridge). See Section 4.4.2 of [RFC6325].

## 7. Processing Address Sets

The following processes should be followed in interpreting sets of AFN values in an IA APPsub-TLV to synthesize addresses. These apply whether the AFN values came from sub-sub-TLVs, appeared within an Address Set, or came from both sources. In general, the processing is applied separately to each Address Set as supplemented by any Fixed Address sub-sub-TLVs that are present.

The OUI AFN value is provided so that MAC addresses can be abbreviated if they have the same upper 24 bits. A MAC/24 is a 24-bit suffix intended to be prefixed by an OUI to create a 48-bit MAC address [RFC7042]; in the absence of an OUI, a MAC/24 entry cannot be used. A MAC/40 is a 40-bit suffix intended to be prefixed by an OUI to create a 64-bit MAC address [RFC7042]; in the absence of an OUI, a MAC/40 entry cannot be used.



Typically, an OUI would be provided as a Fixed Address sub-sub-TLV (see Section 3.2) using the OUI AFN, but there is no prohibition against one or more OUIs appearing in an Address Set.

Each Address Set, after being supplemented by any Fixed Address sub-sub-TLVs, is processed by combining each OUI in the Address Set with each MAC/24 and each MAC/40 address in the Address Set. Depending on how many of each of these address types are present, zero or more 48-bit and/or 64-bit MAC addresses may be synthesized that are subsequently processed as if they had been part of the Address Set. If there are no MAC/24 or MAC/40 addresses present, any OUIs are ignored. If there are no OUIs, any MAC/24s and/or MAC/40s are ignored. If there are K1 OUIs, K2 MAC/24s, and K3 MAC/40s,  $K1 \times K2$  48-bit MACs are synthesized and  $K1 \times K3$  64-bit MACs are synthesized.

IPv6/64 is an 8-byte quantity that is the first 64 bits of an IPv6 address. IPv6/64s are ignored unless, after the processing described above in this subsection, there are one or more 48-bit and/or 64-bit MAC addresses in the Address Set to provide the lower 64 bits of the IPv6 address. For this purpose, a 48-bit MAC address is expanded to 64 bits as described in Section 2.2.1 of [RFC7042]. If there are K4 IPv6/64s present and K5 48-bit and 64-bit MAC addresses present,  $K4 \times K5$  128-bit IPv6 addresses are synthesized.

Synthesized addresses are treated as if they had been members of the Address Set.

## 8. References

### 8.1. Normative References

[ISO-10589]

International Organization for Standardization, "Intermediate System to Intermediate System intra-domain routing information exchange protocol for use in conjunction with the protocol for providing the connectionless-mode network service (ISO 8473)", ISO Standard 10589, 2002.

[RFC826]

Plummer, D., "Ethernet Address Resolution Protocol: Or Converting Network Protocol Addresses to 48.bit Ethernet Address for Transmission on Ethernet Hardware", STD 37, RFC 826, DOI 10.17487/RFC0826, November 1982, <<http://www.rfc-editor.org/info/rfc826>>.

[RFC903]

Finlayson, R., Mann, T., Mogul, J., and M. Theimer, "A Reverse Address Resolution Protocol", STD 38, RFC 903, DOI 10.17487/RFC0903, June 1984, <<http://www.rfc-editor.org/info/rfc903>>.

[RFC2119]

Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.

[RFC4861]

Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, DOI 10.17487/RFC4861, September 2007, <<http://www.rfc-editor.org/info/rfc4861>>.

[RFC5120]

Przygienda, T., Shen, N., and N. Sheth, "M-ISIS: Multi Topology (MT) Routing in Intermediate System to Intermediate Systems (IS-ISs)", RFC 5120, DOI 10.17487/RFC5120, February 2008, <<http://www.rfc-editor.org/info/rfc5120>>.

[RFC5226]

Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, DOI 10.17487/RFC5226, May 2008, <<http://www.rfc-editor.org/info/rfc5226>>.

[RFC5305]

Li, T. and H. Smit, "IS-IS Extensions for Traffic Engineering", RFC 5305, DOI 10.17487/RFC5305, October 2008, <<http://www.rfc-editor.org/info/rfc5305>>.

- [RFC6325] Perlman, R., Eastlake 3rd, D., Dutt, D., Gai, S., and A. Ghanwani, "Routing Bridges (R Bridges): Base Protocol Specification", RFC 6325, DOI 10.17487/RFC6325, July 2011, <<http://www.rfc-editor.org/info/rfc6325>>.
- [RFC6823] Ginsberg, L., Previdi, S., and M. Shand, "Advertising Generic Information in IS-IS", RFC 6823, DOI 10.17487/RFC6823, December 2012, <<http://www.rfc-editor.org/info/rfc6823>>.
- [RFC7042] Eastlake 3rd, D. and J. Abley, "IANA Considerations and IETF Protocol and Documentation Usage for IEEE 802 Parameters", BCP 141, RFC 7042, DOI 10.17487/RFC7042, October 2013, <<http://www.rfc-editor.org/info/rfc7042>>.
- [RFC7172] Eastlake 3rd, D., Zhang, M., Agarwal, P., Perlman, R., and D. Dutt, "Transparent Interconnection of Lots of Links (TRILL): Fine-Grained Labeling", RFC 7172, DOI 10.17487/RFC7172, May 2014, <<http://www.rfc-editor.org/info/rfc7172>>.
- [RFC7356] Ginsberg, L., Previdi, S., and Y. Yang, "IS-IS Flooding Scope Link State PDUs (LSPs)", RFC 7356, DOI 10.17487/RFC7356, September 2014, <<http://www.rfc-editor.org/info/rfc7356>>.
- [RFC7357] Zhai, H., Hu, F., Perlman, R., Eastlake 3rd, D., and O. Stokes, "Transparent Interconnection of Lots of Links (TRILL): End Station Address Distribution Information (ESADI) Protocol", RFC 7357, DOI 10.17487/RFC7357, September 2014, <<http://www.rfc-editor.org/info/rfc7357>>.
- [RFC7780] Eastlake 3rd, D., Zhang, M., Perlman, R., Banerjee, A., Ghanwani, A., and S. Gupta, "Transparent Interconnection of Lots of Links (TRILL): Clarifications, Corrections, and Updates", RFC 7780, DOI 10.17487/RFC7780, February 2016, <<http://www.rfc-editor.org/info/rfc7780>>.

## 8.2. Informative References

- [ARPND] Li, Y., Eastlake 3rd, D., Dunbar, L., and R. Perlman, "TRILL: ARP/ND Optimization", Work in Progress, draft-ietf-trill-arp-optimization-06, April 2016.
- [ChannelTunnel] Eastlake 3rd, D., Umair, M., and Y. Li, "TRILL: RBridge Channel Header Extension", Work in Progress, draft-ietf-trill-channel-tunnel-11, August 2016.
- [DirectoryScheme] Eastlake 3rd, D., Dunbar, L., Perlman, R., and Y. Li, "TRILL: Edge Directory Assist Mechanisms", Work in Progress, draft-ietf-trill-directory-assist-mechanisms-07, February 2016.
- [RFC5494] Arkko, J. and C. Pignataro, "IANA Allocation Guidelines for the Address Resolution Protocol (ARP)", RFC 5494, DOI 10.17487/RFC5494, April 2009, <<http://www.rfc-editor.org/info/rfc5494>>.
- [RFC7067] Dunbar, L., Eastlake 3rd, D., Perlman, R., and I. Gashinsky, "Directory Assistance Problem and High-Level Design Proposal", RFC 7067, DOI 10.17487/RFC7067, November 2013, <<http://www.rfc-editor.org/info/rfc7067>>.
- [RFC7178] Eastlake 3rd, D., Manral, V., Li, Y., Aldrin, S., and D. Ward, "Transparent Interconnection of Lots of Links (TRILL): RBridge Channel Support", RFC 7178, DOI 10.17487/RFC7178, May 2014, <<http://www.rfc-editor.org/info/rfc7178>>.

## Appendix A. Examples

Below are example IA APPsub-TLVs. "0x" indicates that the quantity is in hexadecimal. "0b" indicates that the quantity is in binary. Leading zeros are retained.

### A.1. Simple Example

Below is an annotated IA APPsub-TLV carrying two simple pairs of EUI-48 MAC addresses and IPv4 addresses from a Push Directory (a directory conforming to the Push Model [RFC7067]). No sub-sub-TLVs are included.

```
0x0002(10)  Type: Interface Addresses
0x001B      Length: 27 (= 0x1B)
0x001B      Address Sets End: 27 (= 0x1B)
0x1234      RBridge Nickname from which reachable
0b10000000  Flags: Push Directory data
0xE3        Confidence = 227
33          Template: 33 (0x21) = 32 + 1(IPv4)
```

#### Address Set One

```
0x00005E0053A9  48-bit MAC address
198.51.100.23    IPv4 address
```

#### Address Set Two

```
0x00005E00536B  48-bit MAC address
203.0.113.201    IPv4 address
```

The size includes 7 for the fixed fields through and including the 1-byte Template, plus 2 times the Address Set size. Each Address Set is 10 bytes: 6 for the 48-bit MAC address plus 4 for the IPv4 address. Therefore, the total size is  $7 + 2 \times 10 = 27$ .

See Section 2 for more information on the Template.

## A.2. Complex Example

Below is an annotated IA APPsub-TLV carrying three sets of addresses, each consisting of an EUI-48 MAC address, an IPv4 address, an IPv6 address, and an RBridge Port ID, all from a Push Directory (a directory conforming to the Push Model [RFC7067]). The IPv6 address for each Address Set is synthesized from the MAC address given in that set and the IPv6/64 64-bit prefix provided through a Fixed Address sub-sub-TLV. In addition, a sub-sub-TLV is included that provides an FGL that overrides whatever Data Label may be provided by the envelope (for example, an ESADI-LSP [RFC7357]) within which this IA APPsub-TLV occurs.

```

0x0002(10)   Type: Interface Addresses
0x0036       Length: 64 (= 0x40)
0x0021       Address Sets End: 43 (= 0x2B)
0x4321       RBridge Nickname from which reachable
0b10000000   Flags: Push Directory data
0xD3         Confidence = 211
37           Template: 37(0x25) = 32 + 1(IPv4) + 4(Port)

```

### Address Set One

```

0x00005E0053DE 48-bit MAC address
198.51.100.105  IPv4 address
0x1DE3          RBridge Port ID

```

### Address Set Two

```

0x00005E0053E3 48-bit MAC address
203.0.113.89   IPv4 address
0x1DEE          RBridge Port ID

```

### Address Set Three

```

0x00005E0053D3 48-bit MAC address
192.0.2.139    IPv4 address
0x01DE          RBridge Port ID

```

### sub-sub-TLV One

```

0x0003       Type: Data Label
0x0003       Length: Implies FGL
0xD3E3E3     Fine-Grained Label

```

### sub-sub-TLV Two

```

0x0002       Type: Fixed Address
0x000A       Size: 0x0A = 10
0x400A       AFN: IPv6/64
0x20010db800000000 IPv6 Prefix: 2001:db8::

```

See Section 2 for more information on the Template.

The Fixed Address sub-sub-TLV causes the IPv6/64 value given to be treated as if it occurred as a fourth entry inside each of the three Address Sets. When there is an IPv6/64 entry and a 48-bit MAC entry, the MAC value is expanded by inserting 0xfffe immediately after the OUI, and the local/global bit is inverted. The resulting Modified EUI-64-bit value is used as the lower 64 bits of the resulting IPv6 address (Section 2.2.1 of [RFC7042]). As a result, a receiving TRILL switch would treat the three Address Sets shown as if they had an IPv6 address in them, as follows:

```

      Address Set One
0x20010db80000000002005efffe0053de  IPv6 Address

      Address Set Two
0x20010db80000000002005efffe0053e3  IPv6 Address

      Address Set Three
0x20010db80000000002005efffe0053d3  IPv6 Address

```

As an alternative to the compact "well-known value" Template encoding used in the example above, the less compact explicit AFN encoding could have been used. In that case, the IA APPsub-TLV would have started as follows:

```

0x0002(10)  Type: Interface Addresses
0x003C      Length: 60 (= 0x3C)
0x0027      Address Sets End: 39 (= 0x27)
0x4321      RBridge Nickname from which reachable
0b10000000  Flags: Push Directory data
0xD3        Confidence = 211
0x3         Template: 3 AFNs
0x4005      AFN: 48-bit MAC
0x0001      AFN: IPv4
0x400B      AFN: RBridge Port ID

```

As a final point, since the 48-bit MAC addresses in these three Address Sets all have the same OUI (the IANA OUI [RFC7042]), it would have been possible to just have a MAC/24 value giving the lower 24 bits of the MAC in each Address Set. The OUI would then be supplied by a second Fixed Address sub-sub-TLV providing the OUI. With N Address Sets, this would have saved 3\*N or 9 bytes, at a cost of 9 bytes (2 each for the Type and Length of the sub-sub-TLV, 2 for the OUI AFN, and 3 for the OUI). So, with just three Address Sets, there would be no net savings; however, with a larger number of Address Sets, there would be a net savings.

## Acknowledgments

The authors gratefully acknowledge the contributions and review by the following:

Linda Dunbar, Sue Hares, Paul Kyzivat, Danny McPherson, and Gayle Noble

## Authors' Addresses

Donald Eastlake 3rd  
Huawei Technologies  
155 Beaver Street  
Milford, MA 01757  
United States of America

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

Yizhou Li  
Huawei Technologies  
101 Software Avenue  
Nanjing 210012  
China

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