

TRILL Working Group
INTERNET-DRAFT
Intended status: Proposed Standard

Donald Eastlake
Huawei
Anoop Ghanwani
Dell
Vishwas Manral
HP Networking
Caitlin Bestler
Quantum
June 5, 2012

Expires: December 4, 2012

R Bridges: Further TRILL Header Extensions
<[draft-ietf-trill-rbridge-options-07.txt](#)>

Abstract

The TRILL base protocol standard specifies minimal hooks to safely support TRILL Header extensions. Initial extensions have been specified in RFC [[ExtendHeader](#)]. This document specifies the format for further such extensions and specifies some further specific extensions.

Status of This Memo

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

Distribution of this document is unlimited. Comments should be sent to the TRILL working group mailing list <rbridge@postel.org>.

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

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

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

INTERNET-DRAFT

TRILL Header Options

Table of Contents

1. Introduction.....	3
1.1 Conventions used in this document.....	3
2. TRILL Header Options.....	4
2.1 RBridge Extension Handling Requirements.....	5
2.2 No Critical Surprises.....	6
2.3 Extensions Format.....	6
2.3.1 Flow ID Extension Word.....	7
2.3.2 TLV Extension Format.....	7
2.3.3 Marshaling of Extensions.....	9
2.4 Conflict of Extensions.....	9
3. The ECN Specific Extended Header Flags.....	10
4. Specific TLV Extension.....	12
4.1 Test/Pad Extension.....	12
5. Additions to IS-IS.....	13
6. IANA Considerations.....	13
7. Security Considerations.....	14
8. Acknowledgements.....	14
9. References.....	15
9.1 Normative References.....	15
9.2 Informative References.....	15
Change History.....	16

INTERNET-DRAFT

TRILL Header Options

1. Introduction

The base TRILL protocol standard [[RFC6325](#)] provides a TRILL Header extensions feature, called "options" in [[RFC6325](#)], and describes minimal hooks to safely support header extensions. [[ExtendHeader](#)] extends this by defining the first 32-bit word of the extensions area, which consists of flags, and specifying an initial extended header flag. This draft further specifies the format of and some additional extensions: a special Flow ID field, ECN (Explicit Congestion Notification) extended header flags, and a test/pad extension.

[Section 2](#) below describes the general principles, format, and ordering of TRILL Header Extensions.

[Section 3](#) describes the ECN specific extended flag extensions while [Section 4](#) describes a specific TLV encoded extension.

1.1 Conventions used in this document

The terminology and acronyms defined in [[RFC6325](#)] are used herein with the same meaning.

In this documents, "IP" refers to both IPv4 and IPv6.

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](#)].

[2](#). TRILL Header Options

The base TRILL Protocol includes a feature for extension of the TRILL Header (see [[RFC6325](#)] Sections [3.5](#) and [3.8](#)). The 5-bit Op-Length header field gives the length of the extension to the TRILL Header in units of 4 octets, which allows up to 124 octets of header extension. If Op-Length is zero there is no header extension present; else, this area follows immediately after the Ingress Rbridge Nickname field of the TRILL Header. The optional extensions area starts with a 4-octet word of flags [[ExtendHeader](#)] possibly followed by additional extension words and then TLV extensions. Each TLV extension present is 32-bit aligned.

Provision is also made for both "critical" and "non-critical" extensions. Any RBridge receiving a frame with a critical hop-by-hop extension that it does not implement MUST discard the frame because it is unsafe to process the frame without understanding the critical extension. Any egress RBridge receiving a frame with a critical ingress-to-egress extension it does not implement MUST drop the frame if it is a known unicast frame; if it is a multi-destination TRILL Data frame with a critical ingress-to-egress extension that the RBridge does not implement, then it MUST NOT be egressed at that RBridge but it is still forwarded on the distribution tree. Non-

critical extensions can be safely ignored.

Any extension indicating a significant change in the structure or interpretation of later parts of the frame which, if the extension were ignored, could cause a failure of service or violation of security policy MUST be a critical extension. If such an extension affects any fields that transit R Bridges will examine, it MUST be a hop-by-hop critical extension.

TLV extensions have a "mutability" flag that has a different meaning for hop-by-hop extensions and for extensions other than hop-by-hop.

For extensions other than hop-by-hop, the mutability flag indicates whether the value associated with the extension can change at a transit R Bridge (mutable extensions) or cannot so change (immutable extensions). For example, an ingress-to-egress security extension could protect the value of an immutable ingress-to-egress extension. But such a security extension generally could not protect a mutable value as a transit R Bridge could change that value but might not have the keys to recompute a signature or authentication code to take a changed value into account.

For a non-critical hop-by-hop extension, the mutability flag indicates whether a transit R Bridge that does not implement the extension is permitted (mutable) or not permitted (immutable) to remove the extension. A transit R Bridge is not required to remove a hop-by-hop extension that it does not implement.

For critical hop-by-hop extensions, the mutability flag is meaningless. If the R Bridge does not implement the critical hop-by-hop extension, it MUST drop the frame. If it does implement the critical hop-by-hop extension, it will know whether or not it can remove it. For critical hop-by-hop extensions, the mutability flag is set to zero ("immutable") on transmission and ignored on receipt.

Note: Most R Bridges implementations are expected to be optimized for simple and common cases of frame forwarding and processing. Although the hard limit on the header extensions area length, the 32-bit alignment of TLV extensions, and the presence of critical extension summary bits, as described below, are intended to assist in the efficient hardware based processing of frames with a TRILL header extensions area, nevertheless the inclusion of extensions, particularly TLV extensions, may cause frame processing using a

"slow path" with inferior performance to "fast path" processing. Limited slow path throughput of such frames could cause them to be discarded.

[2.1](#) RBridge Extension Handling Requirements

These requirements are in addition to those in [[ExtendHeader](#)].

All RBridges MUST be able to check whether there are any critical extensions present that are necessarily applicable to their processing of the frame. To assist in this task, critical summary bits are provided that cover all extensions. If an RBridge does not implement all such critical extensions present, it MUST discard the frame or, in some circumstances as described above for certain multi-destination frames, continue to forward the frame but MUST NOT egress the frame.

Transit RBridges MUST be transparent to all immutable ingress-to-egress and immutable reserved header extensions in frames that they forward. Any changes made by a transit RBridge to a mutable ingress-to-egress or reserved extension value MUST be a change permitted by the specification of that extension.

In addition, a transit RBridge:

- o MAY add, if space is available, or remove hop-by-hop extensions as specified for such extensions;
- o MAY change the value and/or length of a mutable ingress-to-egress or reserved TLV extension as permitted by that extension's specification and provided there is enough room if lengthening it;
- o MUST adjust the length of the extensions area, including changing Op-Length in the TRILL header, as appropriate for any changes it has made;

- o MUST NOT add, remove, or re-order ingress-to-egress or reserved extensions.
- o with regard to any non-critical hop-by-hop extensions that the transit RBridge does not implement, it MAY remove them if they are mutable but MUST transparently copy them when forwarding a frame if they are immutable.

[2.2](#) No Critical Surprises

RBridges advertise the ingress-to-egress and reserved extensions they support in IS-IS PDUs [[RFC6326bis](#)] [[MoreISIS](#)] and advertise the hop-by-hop extensions they support at a port on the link connected to that port [[MoreISIS](#)]. An RBridge is not required to support any extensions.

Unless an RBridge advertises support for a critical extension, it will not normally receive frames with that extension.

An RBridge SHOULD NOT add a critical extension to a frame unless,

- for a critical hop-by-hop extensions, it has determined that the next hop RBridge or RBridges that will accept the frame support that extension,
- for a critical ingress-to-egress extensions, it has determined that the RBridge or RBridges that will egress the frame support that extension, or
- for a critical reserved extensions, it may add such an extension only if it understands which RBridges it is applicable to and has determined that those RBridges that will accept the frame support that extension.

"SHOULD NOT" is specified since there may be cases where it is acceptable for those frames, particularly for the multi-destination case, to be discarded by any RBridges that do not implement the extension.

[2.3](#) Extensions Format

If any extensions are present in a TRILL Header, as indicated by a non-zero Op-Length field, the first 32 bits of the extensions area consist of extended header flags as specified in [[ExtendHeader](#)]. The remainder of the extensions area, if any, after this initial 32, consists of one extension word including a Flow ID field and then TLV (Type Length Value) extensions aligned on 32-bit boundaries. [Section 2.3.2](#) specifies the format of a TLV extension. [Section 2.3.3](#) describes the marshaling of TLV extensions.

2.3.1 Flow ID Extension Word

In connection with the multi-pathing of frames, frames that are part of the same order-dependent flow need to follow the same path. Methods to determine flows are beyond the scope of the this document; however, once the flow of a unicast frame has been determined, it can be preserved and transmitted for use by subsequent R Bridges.

A second extension word contains a Flow ID field is present if the extension length TRILL Header field is 2 or larger. It is formatted as follows:

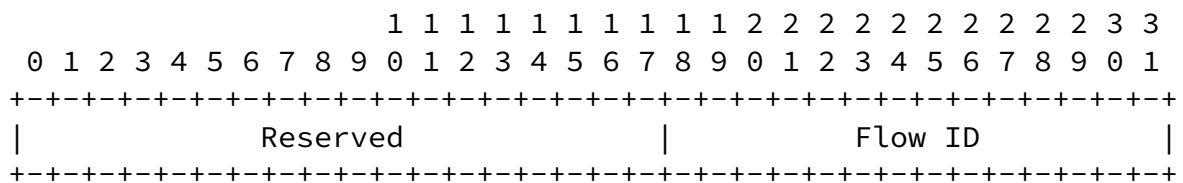


Figure 1. Flow ID Extension Word

The Reserved field in this extension word MUST be sent to zero, transparently copied by transit R Bridges, and ignored on receipt by all R Bridges.

The Flow ID can be considered a special hop-by-hop non-critical option. It can be used for make ECMP forwarding decisions at any transit RBridge. Because the ingress RBridge may know the most about a frame, it is expected that this extension would most commonly be added at the ingress RBridge but if not present, any transit RBridge may add this extension.

When the Flow ID extension word is added, a preceding flags extension word [[ExtendHeader](#)] must also be added.

2.3.2 TLV Extension Format

TRILL Header extensions, other than the extended header flags and Flow ID extension word, are TLV encoded, with some flag bits in the Type and Length octets, in the format show in Figure 2.

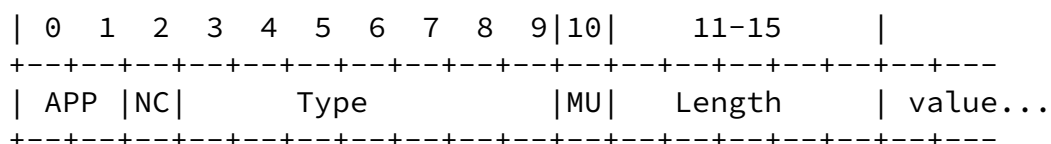


Figure 2. TLV Extension Structure

The APP field gives the applicability of the TLV as follows:

INTERNET-DRAFT

TRILL Header Options

APP Description =====

- 0 Hop-by-hop: Extensions that are potentially applicable to every RBridge that receives the frame.
- 1 Reserved1: Reserved for a class of RBridges to be specified.
- 2 Reserved2: Reserved for a class of RBridges to be specified.
- 3 Ingress-to-egress: Extensions that are only inserted at the ingress and applicable at egress RBridges. Ingress-to-egress extensions MAY also be examined and acted upon by transit RBridges as specified in the particular extension.

For example, Reserved1 might in the future be specified to indicate extensions applicable to multi-level IS-IS border RBridges [[MultiLevel](#)] and Reserved2 to both border and egress RBridges.

Bit 2 in the first octet (NC) is zero for critical extensions and one for non-critical extensions.

Bit 10 in the second octet (MU) is zero for immutable extensions and one for mutable extensions. The APP, NC, Type, and MU fields themselves MUST NOT be changed even for a mutable extension.

The seven-bit Type code extends from bit 3 through bit 9. The extension Type may constrain the values of the APP, NC, and MU bits. For example, a certain Type might require that the extension be marked as a hop-by-hop, non-critical, mutable extension. If the APP, NC, or MU bits have a value not permitted by the extension Type specification for an extension that an RBridge must act on, the RBridge MUST discard the frame. If these bits have a value not permitted by the Type for an extension that an RBridge may ignore, the RBridge MAY discard the frame. "MAY" is chosen in this case to minimize the checking burden.

The Length field is an unsigned quantity giving the length of the extension value in units of four octets. It gives the size of the extension including the initial two Type and Length octets. The Length field MUST NOT be such that the extension value extends beyond the end of the total extensions area as specified by the TRILL Header

Op-Length. Thus, the value 31 is reserved and, when such a value is noticed in a frame, the frame MUST be discarded.

[2.3.3](#) Marshaling of Extensions

In a TRILL Header with extensions, those extensions start immediately after the Ingress RBridge Nickname and fill the extensions area. TLV extensions are 32-bit aligned.

TLV extensions start immediately after the initial four octets of extended flags area [[ExtendHeader](#)] and the Flow ID extensions word (see [Section 2.3.1](#)) and MUST appear in ascending order by the value of the eleven high order bits (bits 0 through 10) of the Type and Length octets considered as an unsigned integer in network byte order. There MUST NOT be more than one extension in a frame with any particular value of this eleven high order bits. Frames that violate this paragraph are erroneous, will produce unspecified results, and MAY be discarded. "MAY" is chosen to minimize the format-checking burden on transit RBridges.

[2.4](#) Conflict of Extensions

It is possible for extensions to conflict. Two or more extensions can be present in a frame that direct an RBridge processing the frame to do conflicting things or to change its interpretation of later parts of the frame in conflicting ways. Such conflicts are resolved by applying the following rules in the order given:

1. Any frame containing extensions that require mutually incompatible changes in way later parts of the frame, after the extensions area, are interpreted or structured MUST be discarded. (Such extensions will be critical extensions, normally hop-by-hop critical extensions.)
2. Critical extensions override non-critical extensions.

2. Within each of the two categories of critical and non-critical extensions, the extension appearing first in lexical order in the frame always overrides an extension appearing later in the frame. For example a conflict between an extended flag and a TLV extension is always resolved in favor of the extended flag.

[3.](#) The ECN Specific Extended Header Flags

RBridges MAY implement an ECN (Explicit Congestion Notification) extension [[RFC3168](#)]. If implemented, it SHOULD be enabled by default but can be disabled on a per RBridge basis by configuration.

RBridges that do not implement this extension or on which it is disabled simply (1) set bits 12 and 13 of the extended flags area to zero when they add an extensions area to a TRILL Header and (2) transparently copy those bits, if an extensions area is present, when they forward a frame with a TRILL Header.

An RBridge that implements the ECN extension does the following, which correspond to the recommended provisions of [[RFC6040](#)], when that extension is enabled:

- o When ingressing an IP frame that is ECN enabled (non-zero ECN field), it MUST add an extensions area to the TRILL Header and copy the two ECN bits from the IP header into extended header flags 12 and 13.
- o When ingressing a frame for a non-IP protocol, where that protocol has a means of indicating ECN that is understood by the RBridge, it MAY add an extensions area to the TRILL Header with the ECN bits set from the ingressed frame.
- o When forwarding a frame encountering congestion at an RBridge, if an extensions area is present with extended flags 12 and 13

- indicating ECN-capable transport, the RBridge MUST modify them to the congestion experienced value.
- o When egressing an IP frame, the RBridge MUST set the outgoing native IP frame ECN field to the code point at the intersection of the values for that field in the encapsulated IP frame (row) and the TRILL extended Header ECN field (column) in Table 2 below or drop the frame in the case where the TRILL header indicates congestion experienced but the encapsulated native IP frame indicates a not ECN-capable transport. (Such frame dropping is necessary because IP transport that is not ECN-capable requires dropped frames to sense congestion.)
 - o When egressing a non-IP protocol frame with a means of indicating ECN that is understood by the RBridge, it MAY set the ECN information in the egressed native frame by combining that information in the TRILL extended header and the encapsulated non-IP native frame as specified in Table 2.

The following table is modified from [\[RFC3168\]](#) and shows the meaning of bit values in TRILL Header extended flags 12 and 13, bits 6 and 7 in the IPv4 TOS Byte, and bits 6 and 7 in the IPv6 Traffic Class Octet:

Binary	Meaning
00	Not-ECT (Not ECN-Capable Transport)
01	ECT(1) (ECN-Capable Transport(1))
10	ECT(0) (ECN-Capable Transport(0))
11	CE (Congestion Experienced)

Table 1. ECN Field Bit Combinations

Table 2 below (adapted from [\[RFC6040\]](#)) shows how, at egress, to combine the ECN information in the extended TRILL Header ECN field with the ECN information in an encapsulated frame to produce the ECN information to be carried in the resulting native frame.

+-----+-----+-----+-----+-----+					
Inner		Arriving TRILL Header ECN Field			
Native		+-----+-----+-----+-----+			
Header		Not-ECT	ECT(0)	ECT(1)	CE

Not-ECT	Not-ECT	Not-ECT(*)	Not-ECT(*)	<drop>(*)
ECT(0)	ECT(0)	ECT(0)	ECT(1)	CE
ECT(1)	ECT(1)	ECT(1)(*)	ECT(1)	CE
CE	CE	CE	CE(*)	CE

Table 2: Egress ECN Behavior

An RBridge detects congestion either by monitoring its own queue depths or from participation in a link-specific protocol. An RBridge implementing the ECN extension MAY be configured to add congestion experienced marking using ECN to any frame with a TRILL Header that encounters congestion even if the frame was not previously marked as ECN-capable or did not have an extensions area.

4. Specific TLV Extension

The table below shows the state of TRILL Header TLV extension Type assignment. See [Section 6](#) for IANA Considerations.

Type	Purpose	Section
0x00	reserved	
0x00-0x3F	available	
0x40	Test/Pad	4.1

0x41-0x7E available
0x7F reserved

Table 3. TLV Extension Types

The following subsection specifies a particular TRILL TLV extension.

[4.1](#) Test/Pad Extension

This extension is intended for testing and padding.

A specific meaning for this extension with the critical flag set will not be defined so, in that form, it **MUST** always be treated as an unknown critical extension. If the critical flag is not set, the extension does nothing. In either case, it may be any length that will fit. Thus, for example, in the non-critical form, it can be used to cause the encapsulated frame starting right after the extensions area to be 64-bit aligned or for testing purposes.

- o Type is 0x40.
- o Length is variable. The value is ignored.
- o IE may be zero or one. This extension has both hop-by-hop and ingress-to-egress versions.
- o NC is zero for the pad extension and one for the test extension.
 - + The non-critical version of this extension does nothing.
 - + The critical version of this extension **MUST** always be treated as an unknown critical extension.
- o MU may be zero or one except that it must be zero if the other flags indicate the extensions is a critical hop-by-hop extension. This extension may be flagged as mutable or immutable.

RBridges use IS-IS PDUs to inform other RBridges which extensions they support. Support for extended header flags is indicated as described in [[RFC6326bis](#)]. The specific IS-IS TLVs or sub-TLVs used to encode and advertise support for TLV options will be specified in a separate document.

[6](#). IANA Considerations

IANA will create a subregistry within the TRILL registry for "TRILL TLV Extension Types" that is initially populated as specified in Table 3 in [Section 4](#). References in that table to sections of this document are to be replaced in the IANA subregistry by references to this document as an RFC.

New TRILL TLV extension types are allocated by IETF Review [[RFC5226](#)].

7. Security Considerations

For general TRILL protocol security considerations, see [[RFC6325](#)].

In order to facilitate authentication, extensions SHOULD be specified so they do not have alternative equivalent forms. Authentication of anything with alternative equivalent forms almost always requires canonicalization that an authenticating RBridge ignorant of the extension would be unable to do and that may be complex and error prone even for an RBridge knowledgeable of the extension. It is best for any extension to have a unique encoding.

8. Acknowledgements

The following are thanked for their contributions: Bob Briscoe.

This document was prepared with basic NROFF. All macros used were defined in the source file.

[9.](#) References

Normative and informative references for this document are given below.

[9.1](#) Normative References

- [RFC2119] - Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [RFC3168] - Ramakrishnan, K., Floyd, S., and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", [RFC 3168](#), September 2001.
- [RFC5226] - Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", [BCP 26](#), [RFC 5226](#), May 2008.
- [RFC6040] - Briscoe, B., "Tunneling of Explicit Congestion Notification", [RFC 6040](#), November 2010
- [RFC6325] - Perlman, R., D. Eastlake, D. Dutt, S. Gai, and A. Ghanwani, "Routing Bridges (RBridges): Base Protocol Specification", July 2011.
- [RFC6326bis] - Eastlake, D., Banerjee, A., Dutt, D., Perlman, R., and A. Ghanwani, "Transparent Interconnection of Lots of Links (TRILL) Use of IS-IS", [draft-eastlake-isis-rfc6326bis](#), work in progress.
- [ExtendHeader] - Eastlake, D., A. Ghanwani, V. Manral, C. Bestler, [draft-ietf-trill-rbridge-extension](#), work in progress.
- [MoreISIS] - tbd

[9.2](#) Informative References

[MultiLevel] - Perlman, R., D. Eastlake, A. Ghanwani, H. Zhai, "RBridges: Multilevel TRILL", [draft-perlman-trill-rbridge-multilevel](#), work in progress.

D. Eastlake, et al

[Page 15]

INTERNET-DRAFT

TRILL Header Options

Change History

The sections below summarize changes between successive versions of this draft. RFC Editor: Please delete this section before publication.

Version 00 to 02

Change the requirement for TLV option ordering to be strictly ordered by the value of the top nine bits of their first two bytes so that the MU bit is included.

Specify meaning of mutability bit for hop-by-hop options.

Fix length of Flow ID Value at 2.

Require that options that may significantly affect the interpretation or format of subsequent parts of the frame be critical options.

Version 02 to 03

Move Test/Pad extension into this document from the More Options draft and move the More Flags option from this document into the More Options draft.

Prohibit multiple occurrences of a TLV option in a frame.

Version 03 to 04

Restructure the bit encoded options area so that the initial 32 bits include a 16-bit Flow ID, various TLV-option-present bits, and a more extended flags bit that means another 32 bits of extended flags are present.

Change the Length of TLV encoded options so that it is in units of 4 bytes, not 1, resulting in a bigger Type field.

Update Explicit Congestion Notification to follow [RFC 6040](#).

Rename "bit encoded options" to be "extended header flags" or "extended flags".

Version 04 to 05

Generally replace "option" with "extension".

Add the Alert critical hop-by-hop flag extension.

Replace MT with MU to avoid possible confusion with multiple topologies.

Version 05 to 06

Update author info.

Update references for issuance of the TRILL base protocol as an RFC.

Remove material now in [[ExtendHeader](#)] and appropriately adjust remaining material including adding references to [[ExtendHeader](#)].

Expand the IE bit in the TLV extension header to the two-bit APP field so as to add the "reserved" type and adjust other material for

the existing of the reserved type of RBridge, different from hop-by-hop and ingress-to-egress.

Version 06 to 07

Update date and version.

D. Eastlake, et al

[Page 17]

INTERNET-DRAFT

TRILL Header Options

Authors' Addresses

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

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

Anoop Ghanwani
Dell
350 Holger Way

San Jose, CA 95134 USA

Phone: +1-408-571-3500

Email: anoop@alumni.duke.edu

Vishwas Manral

HP Networking

19111 Pruneridge Avenue

Cupertino, CA 95014 USA

Tel: +1-408-477-0000

Email: vishwa.manral@hp.com

Caitlin Bestler

Quantum

1650 Technology Drive , Suite 700

San Jose, CA 95110 USA

Phone: +1-408-944-4000

Email: cait@asomi.com

Copyright and IPR Provisions

Copyright (c) 2012 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. The definitive version of an IETF Document is that published by, or under the auspices of, the IETF. Versions of IETF Documents that are published by third parties, including those that are translated into other languages, should not be considered to be definitive versions of IETF Documents. The definitive version of these Legal Provisions is that published by, or under the auspices of, the IETF. Versions of these Legal Provisions that are published by third parties, including those that are translated into other languages, should not be considered to be definitive versions of these Legal Provisions. For the avoidance of doubt, each Contributor to the IETF Standards Process licenses each Contribution that he or she makes as part of the IETF Standards Process to the IETF Trust pursuant to the provisions of [RFC 5378](#). No language to the contrary, or terms, conditions or rights that differ from or are inconsistent with the rights and licenses granted under [RFC 5378](#), shall have any effect and shall be null and void, whether published or posted by such Contributor, or included with or in such Contribution.