TRILL Working Group
INTERNET-DRAFT
Intended status: Proposed Standard

Expires: January 7, 2017

Donald Eastlake
Huawei
Bob Briscoe
Simula Research Lab

July 8, 2016

TRILL: ECN (Explicit Congestion Notification) Support
<draft-eastlake-trill-ecn-support-01.txt>

Abstract

Explicit congestion notification (ECN) allows a forwarding element to notify downstream devices, including the destination, of the onset of congestion without having to drop packets. This document extends this capability to TRILL switches, including integration with IP ECN.

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

Table of Contents

1. Introduction

Explicit congestion notification (ECN [RFC3168]) allows a forwarding element, such as a router, to notify downstream devices, including the destination, of the onset of congestion without having to drop packets. Instead, the forwarding element can explicitly mark a proportion of packets in an ECN field. For example, a two-bit field is available for ECN marking in IP headers.

```
                     .............................
                     .                           .
                 +---------+                     .
    +------+     | Ingress |                     .
    |Source|  +->| RBridge |                     .   +----------+
    +---+--+  |  |   RB1   |                     .   |Forwarding|
        |     |  +------+--+  +----------+       .   | Element  |
        v     |      .  |     | Transit  |       .   |    Y     |
      +-------+--+   .  +---->| RBridges |       .   +--------+-+
      |Forwarding|   .        |   RBn    |       .      ^     |
      | Element  |   .        +-------+--+  +---------+ |     v
      |    X     |   .                |     | Egress  | |  +-----------+
      +----------+   .                +---->| RBridge +-+  |Destination|
                     .                      |   RB9   |    +-----------+
                     .  TRILL               +---------+
                     .  campus                   .
                     .............................
```

Figure 1. Example Path Forwarding Nodes

In [RFC3168] it was recognized that tunnels and lower layer protocols would need to support ECN, and ECN markings would need to be propagated, as headers were encapsulated and decapsulated. [ECNencapGuide] gives guidelines on the addition of ECN to protocols like TRILL that often encapsulate IP packets, including propagation from and to IP.

In the figure above, assuming IP traffic, RB1 is an encapsulator and RB9 a decapsulator. Traffic from Source to RB1 might or might not get marked as having experienced congestion in forwarding elements, such as X, before being encapsulated at ingress RB1. Any such ECN marking is encapsulated with a TRILL Header.

This specification provides for any ECN marking in the traffic at the ingress to be copied into the TRILL Extension Header Flags Word. It also enables a congested transit RBridge such as RBn or RB1 above to introduce congestion marking into the Extension Header Flags Word.

At RB9, the TRILL egress, it specifies how any ECN markings in the TRILL Header Flags Word and in the encapsulated traffic are combined

so that subsequent forwarding elements, such as Y and the Destination, can see if congestion was experienced at any previous point in the path from Source.

A large part of the guidelines for adding ECN to lower layer protocols [ECNencapGuide] concerns safe propagation of congestion notifications in scenarios where some of the nodes do not support or understand ECN. Whichever Rbridges do not support ECN, this specification ensures congestion notification will propagate safely to Destination, as dropped rather than marked packets where necessary.

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].

Acronyms:

AQM - Active Queue Management

CCE - Critical Congestion Experienced

CE - Congestion Experienced

CItE - Critical Ingress-to-Egress

ECN - Explicit Congestion Notification

ECT - ECN Capable Transport

L4S - Low Latency, Low Loss, Scalable throughput

NCHbH - Non-Critical Hop-by-Hop

NCCE - Non-Critical Congestion Experienced

Not-ECT - Not ECN-Capable Transport

PCN - Pre-Congestion Notification

2. The ECN Specific Extended Header Flags

The extension header fields for explicit congestion notification (ECN) in TRILL are defined as a two-bit TRILL-ECN field and a one-bit Critical Congestion Experienced (CCE) field in the TRILL Header Extension Flags Word [RFC7780].

These fields are show in Figure 2 as "ECN" and "CCE". The TRILL-ECN field consists of bits 12 and 13, which are in the range reserved for non-critical hop-by-hop (NCHbH) bits. The CCE field consists of bit 26, which is in the range reserved for Critical Ingress-to-Egress (CItE) bits. See [RFC7780] and [RFC7179] for the meaning of the other bits.

```
    0                   1                   2                   3
```

```
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |Crit.|  CHbH    |   NCHbH    |CRSV | NCRSV |    CItE     |  NCItE   |
    |.....|.........|...........|.....|.......|...........|.........|
    |C|C|C|       |C|N|     |   |     |       |         | |   |     |
    |R|R|R|       |R|C|     |ECN| Ext |       |         |C|Ext|     |
    |H|I|R|       |C|C|     |   | Hop |       |         |C|Clr|     |
    |b|t|s|       |A|A|     |   | Cnt |       |         |E|   |     |
    |H|E|v|       |F|F|     |   |     |       |         | |   |     |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2 The ECN and CCE TRILL Header Extension Flags Word Fields

Table 1 shows the meaning of the codepoints in the TRILL-ECN field. Note that the first three have the same meaning as the corresponding ECN field codepoints in the IPv4 or IPv6 header as defined in [RFC3168]. However codepoint 11 is called Non-Critical Congestion Experienced (NCCE) to distinguish it from Congestion Experienced in IP.

Binary	Name	Meaning
00	Not-ECT	Not ECN-Capable Transport
01	ECT(1)	ECN-Capable Transport (1)
10	ECT(0)	ECN-Capable Transport (0)
11	NCCE	Non-Critical Congestion Experienced

Table 1. TRILL-ECN Field Codepoints

3. ECN Support

The subsections below describe the required behavior to support ECN at TRILL ingress, transit, and egress. The ingress behavior logically occurs as a native frame is encapsulated with a TRILL Header to produce a TRILL Data packet. The transit behavior logically occurs in all RBridges where TRILL Data packets are queues, usually at the output port. The egress behavior logically occurs as a TRILL Data packet is decapsulated and output as a native frame through an RBridge port.

An RBridge that supports ECN MUST behave as described in the relevant subsections below, which correspond to the recommended provisions of [ECNencapGuide]. Nonetheless, the scheme is designed to safely propagate some form of congestion notification even if some RBridges in the path followed by a TRILL Data packet support ECN and others do not.

3.1 Ingress ECN Support

The ingress behavior is as follows:

- o When encapsulating an IP frame, the ingress RBridge MUST:

    - + set the F flag in the main TRILL header [RFC7780];
    - + create a Flags Word as part of the TRILL Header;
    - + copy the two ECN bits from the IP header into the TRILL-ECN field (Flags Word bits 12 and 13)
    - + ensure the CCE flag is cleared to zero (Flags Word bit 26).

- o When encapsulating a frame for a non-IP protocol, where that protocol has a means of indicating ECN that is understood by the ingress RBridge, it MUST follow the guidelines in [ECNencapGuide] to add a Flags Word to the TRILL Header. For a non-IP protocol with a similar ECN field to IP, this would be achieved by copying the TRILL-ECN field from the encapsulated native frame.

3.2 Transit ECN Support

The transit behavior is as follows:

- o An RBridge that supports ECN MUST implement some form of active queue management (AQM) according to the guidelines of [RFC7567]. The RBridge detects congestion either by monitoring its own queue depth or by participating in a link-specific protocol.

o If the TRILL header Flags Word is present, whenever the AQM algorithm decides to indicate congestion on a TRILL Data packet it MUST set the CCE flag (Flags Word bit 26).

o If the TRILL header Flags Word is not present, to indicate congestion the RBridge will either drop the packet or it MAY do all of the following instead:

   + set the F flag in the main TRILL header;
   + add a Flags Word;
   + set the TRILL-ECN field to Not-ECT (00);
   + and set the CCE flag and the Ingress-to-Egress critical summary bit (CRIbE).

Note that a transit RBridge that supports ECN does not refer to the TRILL-ECN field before signalling CCE in a packet. It signals CCE irrespective of whether the packet indicates that the transport is ECN-capable. The egress/decapsulation behavior (described next) ensures that a CCE indication is converted to a drop if the transport is not ECN-capable.

3.3 Egress ECN Support

If the egress RBridge does not support ECN, it will ignore bits 12 and 13 of any Flags Word that is present, because it does not contain any special ECN logic. Nonetheless, if a transit RBridge has set the CCE flag, the egress will drop the packet. This is because drop is the default behavior for an RBridge decapsulating a Critical Ingress-to-Egress flag when it has no specific logic to understand it. Drop is the intended behavior for such a packet, as required by [ECNencapGuide].

If an RBridge supports ECN, the egress behavior is as follows:

o When decapsulating an inner IP packet, the RBridge sets the ECN field of the outgoing native IP packet using Table 2. It MUST set the ECN field of the outgoing IP packet to the codepoint at the intersection of the row for the arriving encapsulated IP packet and the column for 3-bit ECN codepoint in the arriving outer TRILL Data packet header.

   The name of the TRILL 3-bit ECN codepoint is defined using the combination of the TRILL-ECN and CCE fields in Table 3. Specifically, the TRILL 3-bit ECN codepoint is called CE if either NCCE or CCE is set in the TRILL Header Extension Flags Word. Otherwise it has the same name as the 2-bit TRILL-ECN codepoint.

   In the case where the TRILL 3-bit ECN codepoint indicates congestion experienced (CE) but the encapsulated native IP frame indicates a not ECN-capable transport (Not-ECT), it can be seen that the RBridge MUST drop the packet. Such packet dropping is necessary because a transport above the IP layer that is not ECN-capable will have no ECN logic, so it will only understand dropped packets as an indication of congestion.

o When decapsulating a non-IP protocol frame with a means of indicating ECN that is understood by the RBridge, it MUST follow the guideines in [ECNencapGuide] when setting the ECN information in the decapsulated native frame. For a non-IP protocol with a similar ECN field to IP, this would be achieved by combining the information in the TRILL Header flags word with the encapsulated non-IP native frame, as specified in Table 2.

Inner Native Header	Arriving TRILL 3-bit ECN Codepoint Name			
	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 asterisk in the above table indicates a currently unused combination that SHOULD be logged. In contrast to [RFC6040], in TRILL the drop condition is the result of a valid combination of events and need not be logged.

TRILL-ECN	CCE	Arriving TRILL 3-bit ECN codepoint name
Not-ECT 00	0	Not-ECT
ECT(1) 01	0	ECT(1)
ECT(0) 10	0	ECT(0)
NCCE 11	0	CE
Not-ECT 00	1	CE
ECT(1) 01	1	CE
ECT(0) 10	1	CE

```
           | NCCE    11 |  1  | CE                     |
           +------------+-----+---------------------+
```

Table 3: Mapping of TRILL-ECN and CCE Fields to TRILL 3-bit ECN Codepoint Name

4. TRILL Support for ECN Variants

This section is informative, not normative.

Section 3 specifies interworking between TRILL and the original standardized form of ECN in IP [RFC3168].

The ECN wire protocol for TRILL (Section 2) has been designed to support the other known variants of ECN, as detailed below. New variants of ECN will have to comply with the guidelines for defining alternative ECN semantics [RFC4774]. It is expected that the TRILL ECN wire protocol is generic enough to support such potential future variants.

4.1 Pre-Congestion Notification (PCN)

The PCN wire protocol [RFC6660] is recognised by the use of a PCN-compatible Diffserv codepoint in the IP header and a non-zero IP-ECN field. For TRILL or any lower layer protocol, equivalent traffic classification codepoints would have to be defined, but that is outside the scope of the current document.

The PCN wire protocol is similar to ECN, except it indicates congestion with two levels of severity. It uses:

- o 11 (CE) as the most severe, termed the Excess-traffic-marked (ETM) codepoint
- o 01 ECT(1) as a lesser severity level, termed the Threshold-Marked (ThM) codepoint.

To implement PCN on a transit RBridge would require a detailed specification. But in brief:

o the TRILL Critical Congestion Experienced (CCE) flag would be used for the Excess-Traffic-Marked (ETM) codepoint;

o ECT(1) in the TRILL-ECN field would be used for the Threshold-Marked codepoint.

Then the ingress and egress behaviors defined in Section 3 would not need to be altered to ensure support for PCN as well as ECN.

4.2 Low Latency, Low Loss, Scalable Throughput (L4S)

L4S is currently only a proposal being considered for adoption onto the IETF's experimental track. An outline of how a transit TRILL RBridge would support L4S [ECNL4S] is given in Appendix A.

5. IANA Considerations

IANA is requested to update the TRILL Extended Header Flags registry by replacing the lines for bits 9-13 and for bits 21-26 with the following:

Bits	Purpose	Reference
9-11	available non-critical hop-by-hop flags	
12-13	TRILL-ECN (Explicit Congestion Notification)	[this doc]
21-25	available critical ingress-to-egress flags	
26	Critical Congestion Experienced (CCE)	[this doc]

6. Security Considerations

TBD

For ECN tunneling security considerations, see [RFC6040].

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

7. Acknowledgements

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

7. Normative References

[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>.

[RFC3168] - Ramakrishnan, K., Floyd, S., and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", RFC 3168, DOI 10.17487/RFC3168, September 2001, <http://www.rfc-editor.org/info/rfc3168>.

[RFC4774] - Floyd, S., "Specifying Alternate Semantics for the Explicit Congestion Notification (ECN) Field", BCP 124, RFC 4774, DOI 10.17487/RFC4774, November 2006, <http://www.rfc-editor.org/info/rfc4774>.

[RFC6325] - Perlman, R., Eastlake 3rd, D., Dutt, D., Gai, S., and A. Ghanwani, "Routing Bridges (RBridges): Base Protocol Specification", RFC 6325, DOI 10.17487/RFC6325, July 2011, <http://www.rfc-editor.org/info/rfc6325>.

[RFC7179] - Eastlake 3rd, D., Ghanwani, A., Manral, V., Li, Y., and C. Bestler, "Transparent Interconnection of Lots of Links (TRILL): Header Extension", RFC 7179, DOI 10.17487/RFC7179, May 2014, <http://www.rfc-editor.org/info/rfc7179>.

[RFC7567] - Baker, F., Ed., and G. Fairhurst, Ed., "IETF Recommendations Regarding Active Queue Management", BCP 197, RFC 7567, DOI 10.17487/RFC7567, July 2015, <http://www.rfc-editor.org/info/rfc7567>.

[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>.

[ECNencapGuide] - B. Briscoe, J. Kaippallimalil, P. Thaler, "Guidelines for Adding Congestion Notification to Protocols that Encapsulate IP", draft-ietf-tsvwg-ecn-encap-guidelines, work in progress.

Informative References

[RFC6040] - Briscoe, B., "Tunnelling of Explicit Congestion Notification", RFC 6040, DOI 10.17487/RFC6040, November 2010, <http://www.rfc-editor.org/info/rfc6040>.

[RFC6660] - Briscoe, B., Moncaster, T., and M. Menth, "Encoding Three Pre-Congestion Notification (PCN) States in the IP Header Using a Single Diffserv Codepoint (DSCP)", RFC 6660, DOI 10.17487/RFC6660, July 2012, <http://www.rfc-editor.org/info/rfc6660>.

[ECNL4S] - K. De Schepper, B. Briscoe, I. Tsang, "Identifying Modified Explicit Congestion Notification (ECN) Semantics for Ultra-Low Queueing Delay", draft-briscoe-tsvwg-ecn-l4s-id, work in progress.

Appendix A. TRILL Transit RBridge Behavior to Support L4S

An initial specification of the Low Latency, Low Loss, Scalable throughput (L4S) wire protocol for IP is given in [ECNL4S]. It is similar to the original ECN wire protoocl for IP [RFC3168], except:

- o An AQM that supports L4S classifies packets with ECT(1) or CE in the IP header into an L4S queue and a "Classic" queue otherwise.

- o the meaning of CE markings applied by an L4S queue is not the same as the meaning of a drop by a "Classic" queue (contrary to the original requirement for ECN [RFC3168]). Instead the likelihood that the Classic queue drops packets is defined as the square of the likelihood that the L4S queue marks packets (e.g. when there is a drop probability of 0.0009 (0.09%) the L4S marking probability will be 0.03 (3%)).

This seems to present a problem for the way that a transit TRILL RBridge defers the choice between marking and dropping to the egress. Nonetheless, the following pseudocode outlines how a transit TRILL RBridge can implement L4S marking in such a way that the egress behavior already described in Section 3.3 for Classic ECN [RFC3168] will produce the desired outcome.

```
/* p is an internal variable calculated by any L4S AQM
 * dependent on the delay being experienced in the Classic queue.
 */

% Classic Queue on TRILL transit
if (p > max(random(), random()) )
   mark(CCE)                              % likelihood: p^2

% L4S Queue on TRILL transit
if (p > max(random()) ) {
   if (p > max(random()) )
      mark(CCE)                           % likelihood: p^2
   else
      mark(NCCE)                          % likelihood: p - p^2
```

}

With the above transit behavior, an egress that supports ECN (Section 3.3) will drop packets or propagate their ECN markings depending on whether the arriving inner header is from a non-ECN-capable or ECN-capable transport.

Even if an egress has no L4S-specific logic of its own, it will drop packets with the square of the probability that an egress would if it did support ECN, for the following reasons:

- o Egress with ECN support:

  - + L4S: propagates both the Critical and Non-Critical CE marks (CCE & NCCE) as a CE mark.

    Likelihood: p^2 + p - p^2 = p

  - + Classic: Propagates CCE marks as CE or drop, depending on inner.

    Likelihood: p^2

- o Egress without ECN support:

  - + L4S: does not propagate NCCE as a CE mark, but drops CCE marks.

    Likelihood: p^2

  - + Classic: drops CCE marks.

    Likelihood: p^2

Authors' Addresses

Donald E. Eastlake, 3rd
Huawei Technologies
155 Beaver Street
Milford, MA 01757 USA

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

Bob Briscoe (editor)
Simula Research Lab

Email: ietf@bobbriscoe.net
URI: http://bobbriscoe.net/

Copyright and IPR Provisions

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. 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.