TCP Maintenance and Minor Extensions (tcpm)
Internet-Draft
Intended status: Experimental
Expires: December 22, 2013

M. Kuehlewind, Ed.
University of Stuttgart
R. Scheffenegger
NetApp, Inc.

June 20, 2013

More Accurate ECN Feedback in TCP
draft-kuehlewind-tcpm-accurate-ecn-02

Abstract

Explicit Congestion Notification (ECN) is an IP/TCP mechanism where network nodes can mark IP packets instead of dropping them to indicate congestion to the end-points. ECN-capable receivers will feedback this information to the sender. ECN is specified for TCP in such a way that only one feedback signal can be transmitted per Round-Trip Time (RTT). Recently, new TCP mechanisms like ConEx or DCTCP need more accurate ECN feedback information in the case where more than one marking is received in one RTT. This document specifies a different scheme for the ECN feedback in the TCP header to provide more than one feedback signal per RTT. Furthermore this document specifies a re-use of the Urgent Pointer in the TCP header if the URG flag is not set to increase the robustness of the proposed ECN feedback scheme.

Table of Contents

1. Introduction

Explicit Congestion Notification (ECN) [RFC3168] is an IP/TCP mechanism where network nodes can mark IP packets instead of dropping them to indicate congestion to the end-points. ECN-capable receivers will feedback this information to the sender. ECN is specified for TCP in such a way that only one feedback signal can be transmitted per Round-Trip Time (RTT). Recently, proposed mechanisms like Congestion Exposure (ConEx) or DCTCP [Ali10] need more accurate ECN feedback information in case when more than one marking is received in one RTT.

This documents specifies a different scheme for the ECN feedback in the TCP header to provide more than one feedback signal per RTT. This modification does not obsolete [RFC3168]. To avoid confusion we call the ECN specification of [RFC3168] 'classic ECN' in this document. This document provides an extension that requires additional negotiation in the TCP handshake by using the TCP nonce sum (NS) bit, as specified in [RFC3540], which is currently not used when SYN is set. If the more accurate ECN extension has been negotiated successfully, the meaning of ECN TCP bits and the ECN NS bit is different from the specification in [RFC3168], as well as some bits of the largely unused TCP Urgent field as long as the URG flag is not set. This document specifies the additional negotiation as well as the new coding of the TCP ECN/NS bits.

The proposed coding scheme maintains the given bit space in the TCP header as the ECN feedback information is needed in a timely manner and as such should be reported in every ACK. The reuse will avoid additional network load as the ACK size or the number of ACKs will not increase. Moreover, the more accurate ECN information will replace the classic ECN feedback if negotiated. Thus those bits are not needed otherwise. But the proposed schemes requires also the use of the NS bit in the TCP handshake as well as for the more accurate ECN feedback. The proposed more accurate ECN feedback extension includes the ECN-Nonce integrity mechanism as some coding space is left open.

1.1. Overview ECN and ECN Nonce in IP/TCP

ECN requires two bits in the IP header. The ECN capability of a packet is indicated when either one of the two bits is set. An ECN sender can set one or the other bit to indicate an ECN-capable transport (ECT) which results in two signals, ECT(0) and ECT(1). A network node can set both bits simultaneously when it experiences congestion. When both bits are set the packet is regarded as "Congestion Experienced" (CE).

In the TCP header the first two bits in byte 14 are defined for the use of ECN. The TCP mechanism for signaling the reception of a congestion mark uses the ECN-Echo (ECE) flag in the TCP header. To enable the TCP receiver to determine when to stop setting the ECN-Echo flag, the CWR flag is set by the sender upon reception of the feedback signal. This leads always to a full RTT of ACKs with ECE set. Thus any additional CE markings arriving within this RTT can not signaled back anymore.

ECN-Nonce [RFC3540] is an optional addition to ECN that is used to protect the TCP sender against accidental or malicious concealment of marked or dropped packets. This addition defines the last bit of

byte 13 in the TCP header as the Nonce Sum (NS) bit. With ECN-Nonce a nonce sum is maintain that counts the occurrence of ECT(1) packets.

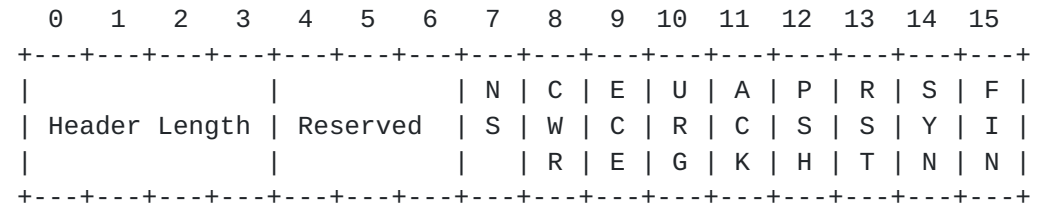

Figure 1: The (post-ECN Nonce) definition of the TCP header flags

1.2. Re-Use of the Urgent field in TCP

RFC0793 specified a mechanism to indicate "urgent data" to a receiver. However, this mechanism is rarely used, and RFC6093 argues to deprecate the use of the mechanism. Furthermore, the content of the Urgent Pointer was always defined to be valid only, when the URG TCP header flag is set. The position of the Urgent Pointer field as well as the URG flag are displayed in Figure 2.

In this document the Urgent Pointer field is defined to be (re)usable for auxiliary data if the URG flag is not set. Note that as the contents of this field were previously undefined when the URG bit is not set, a new mechanism using these bits SHOULD not rely on the correct delivery. Further below in this document a new usage for four bits of the Urgent Pointer counter is defined.

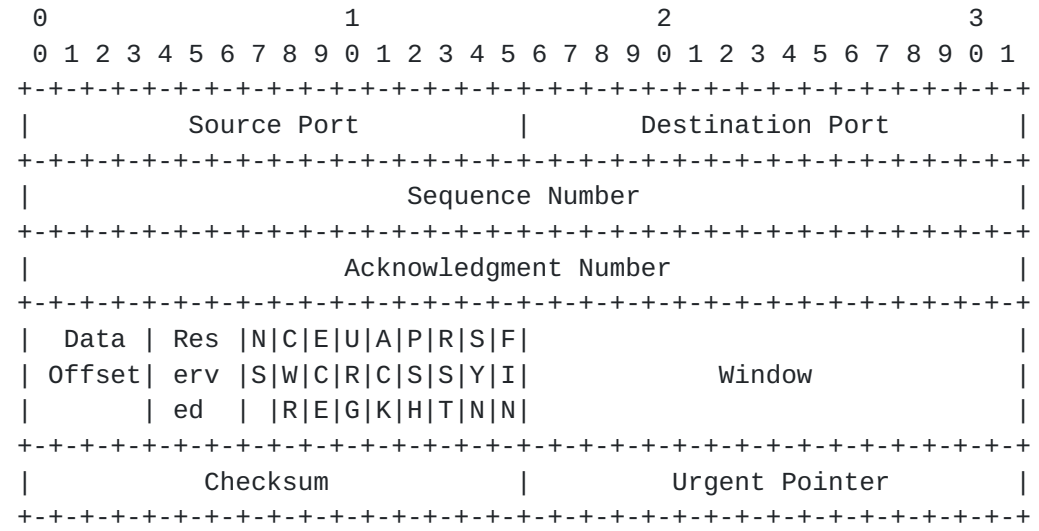

Figure 2: TCP Header Format showing the 16 bit Urgent pointer

1.3. Requirements Language

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].

We use the following terminology from [RFC3168] and [RFC3540]:

The ECN field in the IP header:

CE: the Congestion Experienced codepoint, and

ECT(0): the first ECN-Capable Transport codepoint, and

ECT(1): the second ECN-Capable Transport codepoint.

The ECN flags in the TCP header:

CWR: the Congestion Window Reduced flag,

ECE: the ECN-Echo flag, and

NS: ECN Nonce Sum.

In this document, we will call the ECN feedback scheme as specified in [RFC3168] the 'classic ECN' and our new proposal the 'accurate ECN feedback' scheme. A 'congestion mark' is defined as an IP packet where the CE codepoint is set. A 'congestion event' refers to one or more congestion marks belong to the same overload situation in the network (usually during one RTT).

2. More Accurate ECN Feedback

In this section we designate the sender to be the one sending data and the receiver as the one that will acknowledge this data. Of course such a scenario is describing only one half connection of a TCP connection. The proposed scheme, if negotiated, will be used for both half connection as both, sender and receiver, need to be capable to echo and understand the accurate ECN feedback scheme.

2.1. Negotiation during the TCP handshake

During the TCP handshake at the start of a connection, an originator of the connection (host A) MUST indicate a request to get more accurate ECN feedback by setting the TCP flags NS=1, CWR=1 and ECE=1 in the initial <SYN>. This coding allows to negotiate for the

classic ECN implicit if the receiver does not support the more accurate ECN feedback scheme.

A responding host (host B) MUST return a <SYN,ACK> with flags CWR=1 and ECE=0. The NS flag may be either 0 or 1, as described below. The responding host MUST NOT set this combination of flags unless the preceding <SYN> has already requested support for accurate ECN feedback as above.

These handshakes including the fallback when the receiver only support the classic ECN or ECN-Nonce are summarized in Table 1 below. X indicates that NS can be either 0 or 1 depending on whether congestion had been experienced (see below). The handshake indicating any of the other flavors of ECN are also shown for comparison. To compress the width of the table, the headings of the first four columns have been severely abbreviated, as following:

Ac: *Ac*curate ECN Feedback

N: ECN-*N*once (RFC3540)

E: *E*CN (RFC3168)

I: Not-ECN (*I*mplicit congestion notification).

Ac	N	E	I	<SYN> A->B	<SYN,ACK> B->A	Mode
				NS CWR ECE	NS CWR ECE	
AB				1 1 1	X 1 0	accurate ECN
A	B			1 1 1	1 0 1	ECN Nonce
A		B		1 1 1	0 0 1	classic ECN
A			B	1 1 1	0 0 0	Not ECN
A			B	1 1 1	X 1 1	Not ECN (broken)

Table 1: ECN capability negotiation between Sender (A) and Receiver (B)

The responding host (B) MAY set the NS bit to 1 to indicate a congestion feedback for the <SYN> packet. Otherwise the receiver (B) MUST reply to the sender with NS=0. The addition of ECN to TCP <SYN,ACK> packets is discussed and specified as experimental in [RFC5562] where the addition of ECN to the SYN packet is optionally described. The security implications when using this option are not further discussed here. Only if the initial <SYN> from client A is marked CE, the server B SHOULD set the NS flag to 1 to indicate the congestion immediately, instead of delaying the signal to the first

acknowledgment when the actual data transmission has started. So, server B MAY set the alternative TCP header flags in its <SYN,ACK>: NS=1, CWR=1 and ECE=0.

Recall that, if the <SYN,ACK> reflects the same flag settings as the preceding <SYN> (because there may exist broken TCP implementations that behave this way), [RFC3168] specifies that the whole connection MUST revert to Not-ECT.

2.2. Feedback Coding

This section proposes the new coding to provide a more accurate ECN feedback by use of the two ECN TCP bits (ECE/CWR) as well as the TCP NS bit and the optional use of the Urgent Pointer if the URG flag is not set. This coding MUST only be used if the more accurate ECN Feedback has been negotiated successfully in the TCP handshake.

2.2.1. Codepoint Coding of the more Accurate ECN (ACE) field

The more accurate ECN feedback coding uses the ECE, CWR and NS bits as one field to encode 8 distinct codepoints. This overloaded use of these 3 header flags as one 3-bit more Accurate ECN (ACE) field is shown in Figure 3. The actual definition of the TCP header, including the addition of support for the ECN Nonce, is shown for comparison in Figure 1. This specification does not redefine the names of these three TCP flags, it merely overloads them with another definition once a flow with more accurate ECN feedback is established.

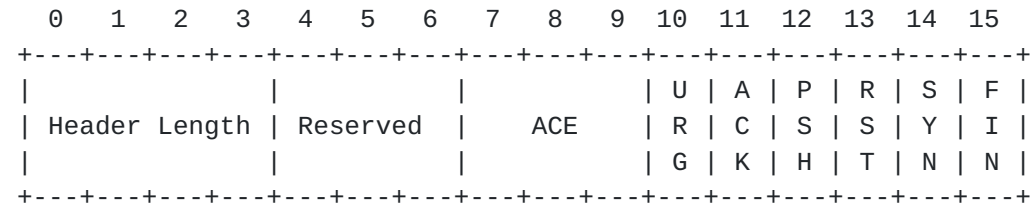

Figure 3: Definition of the ACE field within bytes 13 and 14 of the TCP Header (when SYN=0).

The 8 possible codepoints are shown below. Five of them are used to encode a "congestion indication" (CI) counter. The other three codepoints are defined in the next section to be used for an integrity check based on ECN-Nonce. The CI counter maintains the number of CE marks observed at the receiver (see Section 2.3).

AcE	NS	CWR	ECE	CI (base5)

0	0	0	0	0
1	0	0	1	1
2	0	1	0	2
3	0	1	1	3
4	1	0	0	4
5	1	0	1	-
6	1	1	0	-
7	1	1	1	-

Table 2: Codepoint assignment for accurate ECN feedback

Also note that, whenever the SYN flag of a TCP segment is set (including when the ACK flag is also set), the NS, CWR and ECE flags (i.e. the ACE field of the <SYN,ACK>) MUST NOT be interpreted as the 3-bit codepoint, which is only used in non-SYN packets.

2.2.2. Use with ECN Nonce

In ECN Nonce, by comparing the number of incoming ECT(1) notifications with the actual number of packets that were transmitted with an ECT(1) mark as well as the sum of the sender's two internal counters, the sender can probabilistically detect a receiver that sends false marks or suppresses accurate ECN feedback, or a path that does not properly support ECN.

If an ECT(1) mark is received, an ETC(1) counter (E1) is incremented. The receiver has to convey that updated information to the sender with the next possible ACK using the three remaining codepoints as shown in Table 3.

ECI	NS	CWR	ECE	CI (base5)	E1 (base3)
0	0	0	0	0	-
1	0	0	1	1	-
2	0	1	0	2	-
3	0	1	1	3	-
4	1	0	0	4	-
5	1	0	1	-	0
6	1	1	0	-	1
7	1	1	1	-	2

Table 3: Codepoint assignment for accurate ECN feedback and ECN Nonce

2.2.3. Auxiliary data in the Urgent Pointer field

In order to provide improved resiliency against loss or ACK thinning, the limited number of bits in the existing TCP flags field is insufficient. At the same time is it not necessary to deliver higher order bits with every returned segment, or even reliably at all. Therefore four bits of the reused Urgent Pointer field are defined as the "Top ACE" field of the more accurate ECN feedback, as indicated in Figure 4. This field carries the top (binary) counter value, if the according codepoint does signal the feedback of a counter. Therefore, we call this field "Top ACE".

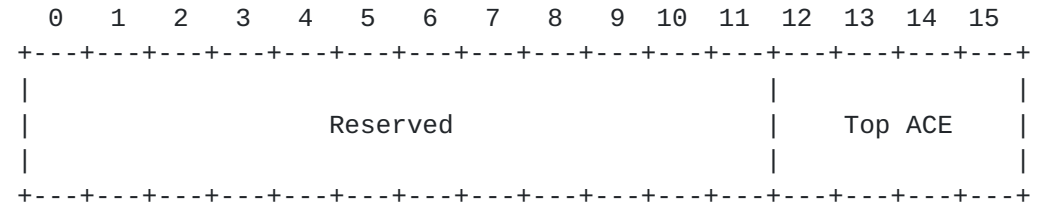

Figure 4: The (post-ECN Nonce) definition of the TCP header flags

As 5 codepoints are set aside to provide reasonable resiliency under typical marking and loss regimes, the combination between the 4 bits in the Top ACE field and the 5 codepoints in the ACE field allow for up to 16*5 = 80 congestion indications to be unambiguously signaled back to the sender, even with more extreme levels of CE marking, or return ACK loss.

A combination with the 3 remaining codepoints (e.g. to signal a counter for the number of observed ECT1 packets) and this field allows for up to 16*3 = 48 distinct indications.

The reserved bits SHOULD be set to zero, and MUST NOT be interpreted when evaluating the combination of the "Top ACE":"ACE" fields. Also, when the URG flag is set, the entire Urgent Pointer MUST NOT be interpreted to carry significance for the Accurate ECN feedback.

2.3. More Accurate ECN TCP Receiver

This section describes the receiver-side action to signal the accurate ECN feedback back to the sender. To select the correct codepoint for each ACK, the receiver will need to maintain a congestion indication (CI) counter of how many CE marking have been seen during a connection and an ECT(1) counter (E1) that is incremented on the reception of a ECT(1) marked packet.

Thus for each incoming segment with a CE marking, the receiver will increase CI by 1. With each ACK the receiver will calculate CI modulo 5 and set the respective codepoint in the ACE field (see Table 2). In addition, the receiver calculates CI divided by 5 and may set

the "Top ACE" field to this value, provided the URG flag is not set in the segment. To avoid counter wrap around in a high congestion situation, the receiver MAY switch from a delayed ACK behavior to send ACKs immediately after the data packet reception if needed.

By default an accurate ECN receiver SHOULD echo the current value of the CI counter, using one of the codepoints encoding the CI counter. Whenever a CE marked segment is received and thus the value of the CI is changed, the receiver MUST echo the then current CI value in the next ACK sent. The receiver MAY use the "Top ACE" field in addition if the URG flag is not set.

The requirement to signal an updated CI value immediately with the next ACK may conflict with a delayed ACK ratios larger than two, when using the available number of codepoints only when "Top ACE" can not be used. A receiver MAY change the ACK'ing rate such that a sufficient rate of feedback signals can be sent. However, in the combination with the redefined Urgent Pointer field, no change in the ACK rate should be required.

Whenever a ECT(1) marked packet arrives, the receiver SHOULD signal the current value of the E1 counter (modulo 3) in the next ACK using the respective codepoint. If a CE mark was received before sending the next ACK (e.g. delayed ACKs) sending the current CI value update MUST take precedence. Further resilience against lost ACKs MAY be provided by inserting the high order bits of the E1 counter (E1 divided by 3) into the Top ACE field.

For the implementation it is suggested to maintain two counters so to avoid costly division operations while processing the header information for the ACK. The first counter can be mapped directly into the ACE field. A wrap by the count of 5 is implemented as a single conditional check, and when that happens, a secondary, high-order counter is increased once. This secondary counter can then be mapped directly into the Top ACE field.

```
if (CE) {
  if (CIcnt == 5) {
    CIcnt = 0
    CIovf += 1
  } else
    CIcnt += 1
}

ACE    = CIcnt;
TopACE = CIovf;
```

Figure 5: Implemetation example

2.4. More Accurate ECN TCP Sender

This section specifies the sender-side action describing how to exclude the number of congestion markings from the given receiver feedback signal.

When the more accurate ECN feedback scheme is supported by the sender, the sender will maintain a congestion indication received (CI.r) counter. This CI.r counter will hold the number of CE marks as signaled by the receiver, and reconstructed by the sender.

On the arrival of every ACK, the sender updates the local CI.r value to the signaled CI value in the ACK as conveyed by the combination of the ACE and "Top ACE" fields in the Urgent Pointer if the URG flag is not set.

If the URG flag is set and thus the "Top ACE" field in the Urgent Pointer field is not available, the sender calculates a value D as the difference between value of the ACE field and the current CI.r value modulo 5. D is assumed to be the number of CE marked packets that arrived at the receiver since it sent the previously received ACK. Thus the local counter CI.r must be increased by D.

As only a limited number of E1 codepoints exist and the receiver might not acknowledge every single data packet immediately (e.g. delayed ACKs), a sender SHOULD NOT mark more than 1/m of the packets with ECT(1), where m is the ACK ratio (e.g. 50% when every second data packet triggers an ACK). This constraint can be lifted when a sender determines, that the auxiliary data is available (the Top ACE field of an ACK with an E1 codepoint is increasing with the number of sent ECT(1) segments). A sender SHOULD send no more than 3 consecutive packets marked with ECT(1), as long as the validity of the auxiliary data in the Top ACE field has not been confirmed.

3. Acknowledgements

We want to thank Bob Briscoe and Michael Welzl for their input and discussion. Special thanks to Bob Briscoe, who first proposed the use of the ECN bits as one field.

4. IANA Considerations

This memo includes a request to IANA, to set up a new registry. This registry redefines the use of the 16 bit "Urgent Pointer" while the URG flag is not set. 4 of those bits ("Top ACE") are defined within this document to be interpreted in conjunction with another field ("ACE"), overwriting three of the existing TCP flags into a single field.

5. Security Considerations

TBD

ACK loss

This scheme sends each codepoint only once. In the worst case at least one, and often two or more consecutive ACKs can be dropped without losing congestion information, even when the auxiliary data field in the former Urgent Pointer field is unavailable (i.e. the URG flag is set, or a middlebox clears its contents).

At low congestion rates, the sending of the current value of the CI counter by default allows higher numbers of consecutive ACKs to be lost, without impacting the accuracy of the ECN signal.

ECN Nonce

In the proposed scheme there are three more codepoints available that could be used for an integrity check like ECN Nonce. If ECN nonce would be implemented as proposed in Section 2.2.2, even more information would be provided for ECN Nonce than in the original specification.

A delayed ACK ratio of two can be sustained indefinitely without reverting to auxiliary information, even during heavy congestion, but not during excessive ECT(1) marking, which is under the control of the sender. A higher ACK ratio can be sustained when congestion is low, and the auxiliary data is available.

6. References

6.1. 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.

[RFC3540] Spring, N., Wetherall, D., and D. Ely, "Robust Explicit Congestion Notification (ECN) Signaling with Nonces", RFC 3540, June 2003.

6.2. Informative References

[Ali10] Alizadeh, M., Greenberg, A., Maltz, D., Padhye, J., Patel, P., Prabhakar, B., Sengupta, S., and M. Sridharan, "DCTCP: Efficient Packet Transport for the Commoditized Data Center", Jan 2010.

[I-D.briscoe-tsvwg-re-ecn-tcp] Briscoe, B., Jacquet, A., Moncaster, T., and A. Smith, "Re-ECN: Adding Accountability for Causing Congestion to TCP/IP", draft-briscoe-tsvwg-re-ecn-tcp-09 (work in progress), October 2010.

[RFC5562] Kuzmanovic, A., Mondal, A., Floyd, S., and K. Ramakrishnan, "Adding Explicit Congestion Notification (ECN) Capability to TCP's SYN/ACK Packets", RFC 5562, June 2009.

[RFC5681] Allman, M., Paxson, V., and E. Blanton, "TCP Congestion Control", RFC 5681, September 2009.

[RFC5690] Floyd, S., Arcia, A., Ros, D., and J. Iyengar, "Adding Acknowledgement Congestion Control to TCP", RFC 5690, February 2010.

[draft-kuehlewind-tcpm-accurate-ecn-option] Kuehlewind, M. and R. Scheffenegger, "Accurate ECN Feedback Option in TCP", draft-kuehlewind-tcpm-accurate-ecn-option-01 (work in progress), Jul 2012.

Authors' Addresses

Mirja Kuehlewind (editor)
University of Stuttgart
Pfaffenwaldring 47
Stuttgart 70569
Germany

Email: mirja.kuehlewind@ikr.uni-stuttgart.de

Richard Scheffenegger
NetApp, Inc.
Am Euro Platz 2
Vienna 1120
Austria

Phone: +43 1 3676811 3146
Email: rs@netapp.com