tcpm	M. Kühlewind, Ed.
Internet-Draft	University of Stuttgart
Intended status: Experimental Protocol	R. Scheffenegger
Expires: April 26, 2012	NetApp, Inc.
	October 24, 2011

Accurate ECN Feedback Option in TCP
draft-kuehlewind-tcpm-accurate-ecn-option-00

Abstract

This document specifies an TCP option to get accurate Explicit Congestion Notification (ECN) feedback from the receiver. ECN is an IP/TCP mechanism where network nodes can mark IP packets instead of dropping them to indicate congestion to the end-points. An ECN-capable receiver 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 feedback information in the case where more than one marking is received in one RTT. This TCP extension can be used in addition to the classic ECN as well as with a more accurate ECN scheme recently proposed which reuses the ECN bit in the TCP header.

Status of this Memo

Copyright Notice

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. An ECN-capable receiver 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 feedback information in case when more than one marking is received in one RTT.
This documents specifies an TCP option to provide more than one ECN feedback signal per RTT. This modification does not obsolete [RFC3168]. This TCP extension can be used in addition to the classic ECN as well as in addition to more accurate ECN scheme recently proposed which reuses the ECN bits in the TCP header for the same purpose than this extension --- more accurate ECN feedback (see [I-D.kuehlewind-conex-accurate-ecn]). Note that a new TCP extension can experience deployment problems by middleboxes dropping unknown options. Thus the ECN feedback in the TCP header is still needed to ensure ECN feedback. Moreover, this option will increase the header length for all kind of TCP packets which causes additional load in case of severe congestion.

1.1. Overview ECN and ECN Nonce in 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 (ETC) which results in two signals --- ECT(0) and respectively ECT(1). A network node can set both bits simultaneously when it experiences congestion. When both bits are set the packets is regarded as "Congestion Experienced" (CE).
ECN-Nonce [RFC3540] is an optional addition to ECN that is used to protects the TCP sender against accidental or malicious concealment of marked or dropped packets. With ECN-Nonce a nonce sum is maintain that counts the occurrence of ECT(1) packets.

1.2. 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)/ECT(1): either one of the two ECN-Capable Transport codepoints.

In this document, we will call the ECN feedback scheme as specified in [RFC3168] the 'classic ECN'. A 'congestion mark' is defined as an IP packet where the CE codepoint is set.

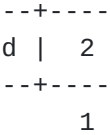
2. Negotiation of Accurate ECN feedback

As there is only limited space in the TCP Options, particularly during the initial three-way handshake, an abbreviated Option is used to negotiate for Accurate ECN feedback. This option also initiates all counters to an initial value of zero at the receiving side.

TCP Accurate ECN Option Negotiation:

Kind: TBD (same as above)

Length: 2 bytes



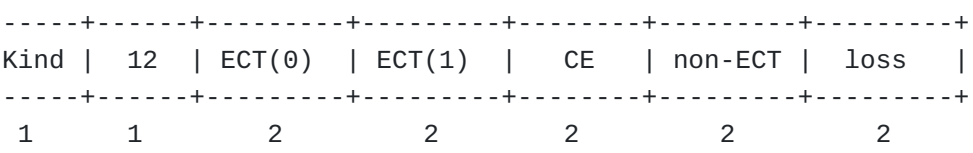
This abbreviated option is only valid in a <SYN> or <SYN,ACK> segment, during a three way handshake. The negotiation follows the same procedure as with other TCP options, i.e. SACK. A TCP sender MAY send the accurate ECN feedback negotiation option in an initial SYN segment and MAY send a more accurate ECN option (see Section 3) in other segments only if it received this option negotiation in the initial <SYN> segment or <SYN,ACK> for the connection. A TCP receiver MAY send an <SYN,ACK> segment with the accurate ECN feedback negotiation option in response to a received accurate ECN feedback negotiation option in the <SYN>. If both ends indicate that they support Accurate ECN feedback, the option MUST be used in any subsequent TCP segment.

3. Accurate ECN feedback Option Specification

TCP Accurate ECN Option:

Kind: TBD

Length: 12 bytes



A TCP receiver, that provides accurate ECN feedback, will maintain a counter for the number of ECT(0), ECT(1), CE, non-ECT marked and lost packets. The TCP option to provide the accurate ECN feedback to the sender will echo these counters (modulo 16) in five 2 byte fields. This option should be included in every ACK (and not only when a counter value changes) to ensure the reception of the ECN feedback at the sender in case of ACK loss.
The number of lost packet would be needed to calculate the ECN Nonce sum more exactly. TCP provides anyway a mechanism to detect loss and lost needs to be assumes as a string signal for congestion anyway. Thus a TCP congestion control algorithm muss react on loss. Moreover, if TCP SACK is not available the exact number of lost packets is not known.
The same feedback information are proposed for the (ECN) feedback in RTP (see [I-D.ietf-avtcore-ecn-for-rtp].
As TCP is a bitdirectional protocol this option can be used in both directions.

4. Acknowledgements

5. IANA Considerations

TBD

6. Security Considerations

TBD

7. References

7.1. 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.

7.2. Informative References

[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", Internet-Draft draft-briscoe-tsvwg-re-ecn-tcp-09, October 2010.
[I-D.ietf-avtcore-ecn-for-rtp]	Westerlund, M, Johansson, I, Perkins, C, O'Hanlon, P and K Carlberg, "Explicit Congestion Notification (ECN) for RTP over UDP", Internet-Draft draft-ietf-avtcore-ecn-for-rtp-05, October 2011.
[I-D.kuehlewind-conex-accurate-ecn]	Kuehlewind, M and R Scheffenegger, "Accurate ECN Feedback in TCP", Internet-Draft draft-kuehlewind-conex-accurate-ecn-01, October 2011.
[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.

Authors' Addresses

Mirja Kühlewind editor Kühlewind University of Stuttgart Pfaffenwaldring 47 Stuttgart, 70569 Germany EMail: mirja.kuehlewind@ikr.uni-stuttgart.de

Richard Scheffenegger Scheffenegger NetApp, Inc. Am Euro Platz 2 Vienna, 1120 Austria Phone: +43 1 3676811 3146 EMail: rs@netapp.com