Network Working Group
Internet-Draft
Updates: 3168,4774 (if approved)
Intended status: Standards Track
Expires: January 21, 2017

N. Khademi
M. Welzl
University of Oslo
G. Armitage
Swinburne University of Technology
G. Fairhurst
University of Aberdeen

July 20, 2016

Updating the Explicit Congestion Notification (ECN) Specification to Allow IETF Experimentation
draft-khademi-tsvwg-ecn-response-01

Abstract

This document relaxes recommendations and prescriptions from RFC3168 and RFC4774 that get in the way of experimentation with different ECN strategies. First, RFC3168 and RFC4774 state that, upon the receipt by an ECN-Capable transport of a single CE packet, the congestion control algorithms followed at the end-systems MUST be essentially the same as the congestion control response to a single dropped packet. This document relaxes this rule in order to encourage experimentation with different backoff strategies. Second, this document allows future IETF specifications to use the ECT(1) codepoint in ways that are currently prohibited by RFC3168. Third, this document allows future IETF experiments to use the ECT(0) or ECT(1) codepoint on any TCP segment.

Table of Contents

1. Introduction

This document relaxes three limitations that are due to specific text in [RFC3168] and, in one case, also [RFC4774].

1.1. Differently reacting to ECN-marks and loss

Explicit Congestion Notification (ECN) as specified in [RFC3168] allows a network device that uses Active Queue Management (AQM) to set the Congestion Experienced (CE) codepoint in the ECN field of the IP packet header, rather than to drop ECN-capable packets when incipient congestion is detected. When an ECN-capable transport is used over a path that supports ECN, this provides the opportunity for flows to improve their performance in the presence of incipient congestion [I-D.AQM-ECN-benefits].

[RFC3168] not only specifies the router use of the ECN field, it also specifies a TCP procedure for using ECN. This states that a TCP sender should treat the ECN indication of congestion in the same way as that of a non-ECN-Capable TCP flow experiencing loss, by halving the congestion window "cwnd" and by reducing the slow start threshold "ssthresh". [RFC5681] stipulates that TCP congestion control sets "ssthresh" to max(FlightSize / 2, 2*SMSS) in response to packet loss. This corresponds to a backoff multiplier of 0.5 (halving cwnd and sshthresh after packet loss). Consequently, a standard TCP flow using this reaction needs significant network queue space: it can only fully utilise a bottleneck when the length of the link queue (or the AQM dropping threshold) is at least the bandwidth-delay product (BDP) of the flow.

A backoff multiplier of 0.5 is not the only available strategy. As defined in [I-D.CUBIC], CUBIC multiplies the current cwnd by 0.7 in response to loss (the Linux implementation of CUBIC has used a multiplier of 0.7 since kernel version 2.6.25 released in 2008). Consequently, CUBIC utilises paths well even when the bottleneck queue is shorter than the bandwidth-delay product of the flow. However, in the case of a DropTail (FIFO) queue without AQM, such less-aggressive backoff increases the risk of creating a standing queue [CODEL2012].

Devices implementing AQM are likely to be the dominant (and possibly only) source of ECN CE-marking for packets from ECN-capable senders. AQM mechanisms typically strive to maintain a small average queue length, regardless of the bandwidth-delay product of flows passing through them. Receipt of an ECN CE-mark might therefore reasonably be taken to indicate that a small bottleneck queue exists in the path, and hence the TCP flow would benefit from using a less aggressive backoff multiplier. Such behavior is however prohibited

by the rules in [RFC3168].

ECN has seen little deployment so far. Apple recently announced their intention to enable ECN in iOS 9 and OS X 10.11 devices [WWDC2015]. By 2014, server-side ECN negotiation was observed to be provided by the majority of the top million web servers [PAM2015], and only 0.5% of websites incurred additional connection setup latency using RFC3168-compliant ECN-fallback mechanisms. [RFC7567] states that "deployed AQM algorithms SHOULD support Explicit Congestion Notification (ECN) as well as loss to signal congestion to endpoints" and [I-D.AQM-ECN-benefits] encourages this deployment. However, the limitation of [RFC3168] restricts a sender to react to notification of a CE-mark in the same way as if a packet was lost. This prohibits experimentation with ECN mechanisms that could yield greater benefits. This specification therefore relaxes this constraint.

1.1.1. Discussion: Why Use ECN to Vary the Degree of Backoff?

The classic rule-of-thumb dictates that a transport provides a BDP of bottleneck buffering if a TCP connection wishes to optimise path utilisation. A single TCP connection running through such a bottleneck will have opened cwnd up to 2*BDP by the time packet loss occurs. [RFC5681]’s halving of cwnd and ssthresh pushes the TCP connection back to allowing only a BDP of packets in flight -- just sufficient to maintain 100% utilisation of the network path.

AQM schemes like CoDel [I-D.CoDel] and PIE [I-D.PIE] use congestion notifications to constrain the queuing delays experienced by packets, rather than in response to impending or actual bottleneck buffer exhaustion. With current default delay targets, CoDel and PIE both effectively emulate a shallow buffered bottleneck (section II, [ABE2015]) while allowing short traffic bursts into the queue. This interacts acceptably for TCP connections over low BDP paths, or highly multiplexed scenarios (many concurrent TCP connections). However, it interacts badly with lightly-multiplexed cases (few concurrent connections) over a high BDP path. Conventional TCP backoff in such cases leads to gaps in packet transmission and under-utilisation of the path.

The idea to react differently to loss upon detecting an ECN CE-mark pre-dates [ABE2015]. [ICC2002] also proposed using ECN CE-marks to modify TCP congestion control behaviour, using a larger multiplicative decrease factor in conjunction with a smaller additive increase factor to work with RED-based bottlenecks that were not necessarily configured to emulate a shallow queue.

This update to [RFC3168] that enables the IETF to specify experiments

with a different backoff behavior in response to a CE-mark than in response to packet loss is utilized by an experiment called "Alternative Backoff with ECN" (ABE). ABE is based upon [ABE2015] and defined in [I-D.ABE].

1.2. Senders setting the ECT(1) codepoint

Future IETF experiments may require setting the ECT(0) or ECT(1) codepoints differently from what [RFC3168] recommends or requires.

[NOTE: This usage was also specified in ECN-NONCE.]

This update may also allow the iETF to specify future mechanisms that associate alternate ECN semantics with this codepoint. An experiment called "L4S" proposes to use the ECT(1) codepoint to indicate in which of two queues a packet should be placed [I-D.briscoe-tsvwg-ecn-l4s-id].

1.3. ECT(0) and ECT(1) on control packets

Diverging from recommendations or requirements in [RFC3168], future IETF experiments may be specified to use the ECT(0) or ECT(1) codepoint. This choice of codepoint can be used to signal alternative ECN semantics. This supersedes the rationale in Section 20 of [RFC3168] that argued against the use of ECT(1) to specify alternate ECN semantics, instead arguing for attaching specific ECN semantics to a Differentiated Services Code Point (DSCP).

This update may also allow the iETF to specify future updates to transport protocol use of ECN. A proposal,[I-D.bagnulo-tsvwg-generalized-ecn], provides arguments for using the ECT(0) or ECT(1) codepoint on a broader range of TCP packets for which such usage is precluded by [RFC3168]: SYNs, pure ACKs, retransmitted packets and window probe packets.

2. Updating RFC3168 and RFC4774

This section specifies updates to [RFC3168] (and corresponding text in [RFC4774]) and refers to experiments that are possible within the framework provided by the update.

2.1. RFC 2119

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].

2.2. Scope of this update

Internet deployment of new mechanisms enabled by this update REQUIRE IETF specification in an Experimental or a Standards Track RFC approved by the IESG.

Some mechanisms rely on ECN semantics that differ from the definitions in [RFC3168] -- for example, Congestion Exposure (ConEx) [RFC7713] and DCTCP [I-D.ietf-tcpm-dctcp] need more accurate ECN information than the feedback mechanism in [RFC3168] offers (defined in [I-D.ietf-tcpm-accurate-ecn]). Such mechanisms allow a sending rate adjustment more frequent than each RTT. These mechanisms are out of the scope of the current document.

The remainder of this section lists a set of changes to [RFC3168] that are not specific replacements of text passages.

2.3. Changes to the meaning of a CE-Mark codepoint

This document specifies an update to the TCP sender reaction that follows when the TCP receiver signals that ECN CE-marked packets have been received.

[RFC3168] and [RFC4774] contain the following text:

"Upon the receipt by an ECN-Capable transport of a single CE packet, the congestion control algorithms followed at the end-systems MUST be essentially the same as the congestion control response to a *single* dropped packet. For example, for ECN-Capable TCP the source TCP is required to halve its congestion window for any window of data containing either a packet drop or an ECN indication."

This memo updates the preceding text by replacing it with the following text:

"Upon the receipt by an ECN-Capable transport of a single CE-Marked packet, the congestion control algorithms followed at the endpoints MUST make a congestion control response as specified in [RFC3168] or its updates. For example, an ECN-Capable TCP sender could halve its congestion window for any window of data containing either a packet drop or an ECN indication."

The first paragraph of Section 6.1.2, "The TCP Sender", in [RFC3168] contains the following text:

"If the sender receives an ECN-Echo (ECE) ACK packet (that is, an ACK packet with the ECN-Echo flag set in the TCP header), then the sender knows that congestion was encountered in the network on the path from

the sender to the receiver. The indication of congestion should be treated just as a congestion loss in non-ECN-Capable TCP. That is, the TCP source halves the congestion window "cwnd" and reduces the slow start threshold "ssthresh"."

This memo updates the preceding text by replacing it with the following text:

"If a TCP sender receives an indication of a receibed ECN-Echo (ECE) ACK packet (that is, an ACK packet with the ECN-Echo flag set in the TCP header), then the sender knows that congestion was encountered in the network on the path from the sender to the receiver. An indication of congestion, signalled by reception of the ECN-Echo flag (with the semantics defined in [RFC3168]) MUST produce a rate reduction of at least 15%, so that flows sharing the same bottleneck can increase their share of the capacity. The indication of congestion could be treated in the same way as if the flow had experienced loss, but future congestion control methods are allowed to specify a reduction that is less than the reduction for congestion loss.

An ECN-capable network device cannot eliminate the possibility of packet loss. A drop may still occur due to a traffic burst exceeding the instantaneous available capacity of a network buffer or as a result of the AQM algorithm (overload protection mechanisms, etc [RFC7567]). Whatever the cause of loss, detection of a missing packet needs to trigger the standard loss-based congestion control response". This update explicitly does not change the use of standard protocol mechanisms following loss, as required in [RFC3168].

2.4. Setting ECT(0) and ECT(1) Codepoints

New IETF specifications MAY permit a sender to set the ECT(0) or ECT(1) codepoint on a protocol control packet (including TCP segments for which [RFC3168] does not allow or recommend setting these codepoints.)

[AUTHORS' NOTE: Future versions of this document may take the form of such explicit text replacements.]

2.5. Clarification to the usage of the ECT(1) Codepoint

[RFC3168] notes that a router may treat and mark/drop packets differently depending on whether they observe the ECT(0) or ECT(1) codepoint. This specification permits new IETF specifications to set or read the ECT(1) codepeoint. It clarifies that these specifications do not necessarily treat ECT(1) as equivalent to

ECT(0).

Network devices using IETF-defined DSCPs MUST NOT re-mark packets to the ECT(1) codepoint. Specifically, the methods described in earlier ECN implementations using this codepoint as a congestion mark (described in Section 11.2.1 of [RFC3168]) are NOT RECOMMENDED for deployment in the current Internet.

3. Acknowledgements

The authors N. Khademi, M. Welzl and G. Fairhurst were part-funded by the European Community under its Seventh Framework Programme through the Reducing Internet Transport Latency (RITE) project (ICT-317700). The views expressed are solely those of the authors.

4. IANA Considerations

XX RFC ED - PLEASE REMOVE THIS SECTION XXX

This memo includes no request to IANA.

5. Security Considerations

The described method is a sender-side only transport change, and does not change the protocol messages exchanged. The security considerations of [RFC3168] therefore still apply.

A congestion control backoff that is less in response to ECN than the response to a packet loss can lead to a change in the capacity achieved when flows share a network bottleneck. This can result in redistribution of capacity between sharing flows, potentially resulting in unfairness in the way that capacity is shared. This potential gain applies only to ECN-marked packets using the updated method (and not to detected packet loss). Similar unfairness can be exhibited by congestion control mechanisms that have been used in the Internet for many years (e.g., CUBIC [I-D.CUBIC]). Unfairness may also be a result of other factors, including the round trip time experienced by a flow.

Packet loss can be expected from an AQM algorithm experiencing persistent queuing, but could also imply the presence of faulty equipment or media in a path, or it may imply the presence of congestion [RFC7567]. The update does not change the congestion control response to packet loss, and will therefore not lead to congestion collapse.

[AUTHORS' NOTE: Security considerations of the more relaxed rules of using ECT(0) vs. ECT(1) and usage of these ECT codepoints on any TCP segments will be included in the next version of this document.]

6. Revision Information

XX RFC ED - PLEASE REMOVE THIS SECTION XXX

-01. Broadened the scope to also cover ECT(0) vs. ECT(1) usage and using ECT(0) or ECT(1) codepoints on any TCP segments.

-00. draft-khademi-tsvwg-ecn-response-00 and draft-khademi-tcpm-alternativebackoff-ecn-00 replace draft-khademi-alternativebackoff-ecn-03, following discussion in the TSVWG and TCPM working groups.

7. References

7.1. 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>.

[RFC5681] Allman, M., Paxson, V., and E. Blanton, "TCP Congestion Control", RFC 5681, DOI 10.17487/RFC5681, September 2009, <http://www.rfc-editor.org/info/rfc5681>.

[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>.

7.2. Informative References

[ABE2015] Khademi, N., Welzl, M., Armitage, G., Kulatunga, C., Ros, D., Fairhurst, G., Gjessing, S., and S. Zander, "Alternative Backoff: Achieving Low Latency and High Throughput with ECN and AQM", CAIA Technical Report CAIA-TR-150710A, Swinburne University of Technology, July 2015, <http://caia.swin.edu.au/reports/150710A/CAIA-TR-150710A.pdf>.

[CODEL2012] Nichols, K. and V. Jacobson, "Controlling Queue Delay", July 2012, <http://queue.acm.org/detail.cfm?id=2209336>.

[I-D.ABE] Khademi, N., Welzl, M., Armitage, G., and G. Fairhurst, "TCP Alternative Backoff with ECN (ABE)", Internet-draft, IETF work-in-progress draft-khademi-tcpm-alternativebackoff-ecn-00, May 2016.

[I-D.AQM-ECN-benefits] Fairhurst, G. and M. Welzl, "The Benefits of using Explicit Congestion Notification (ECN)", Internet-draft, IETF work-in-progress draft-ietf-aqm-ecn-benefits-08, November 2015.

[I-D.CUBIC] Rhee, I., Xu, L., Ha, S., Zimmermann, A., Eggert, L., and R. Scheffenegger, "CUBIC for Fast Long-Distance Networks", Internet-draft, IETF work-in-progress draft-ietf-tcpm-cubic-01, January 2016.

[I-D.CoDel] Nichols, K., Jacobson, V., McGregor, V., and J. Iyengar, "Controlled Delay Active Queue Management", Internet-draft, IETF work-in-progress draft-ietf-aqm-codel-03, March 2016.

[I-D.PIE] Pan, R., Natarajan, P., Baker, F., White, G., VerSteeg, B., Prabhu, M., Piglione, C., and V. Subramanian, "PIE: A Lightweight Control Scheme To Address the Bufferbloat Problem", Internet-draft, IETF work-in-progress draft-ietf-aqm-pie-07, April 2016.

[I-D.bagnulo-tsvwg-generalized-ecn] Bagnulo, M. and B. Briscoe, "Adding Explicit Congestion Notification (ECN) to TCP control packets", draft-bagnulo-tsvwg-generalized-ecn-01 (work in progress), July 2016.

[I-D.briscoe-tsvwg-ecn-l4s-id]
Schepper, K., Briscoe, B., and I. Tsang, "Identifying Modified Explicit Congestion Notification (ECN) Semantics for Ultra-Low Queuing Delay", draft-briscoe-tsvwg-ecn-l4s-id-01 (work in progress), March 2016.

[I-D.ietf-tcpm-accurate-ecn]
Briscoe, B., Kuehlewind, M., and R. Scheffenegger, "More Accurate ECN Feedback in TCP", draft-ietf-tcpm-accurate-ecn-00 (work in progress), December 2015.

[I-D.ietf-tcpm-dctcp]
Bensley, S., Eggert, L., Thaler, D., Balasubramanian, P., and G. Judd, "Datacenter TCP (DCTCP): TCP Congestion Control for Datacenters", draft-ietf-tcpm-dctcp-01 (work in progress), November 2015.

[ICC2002] Kwon, M. and S. Fahmy, "TCP Increase/Decrease Behavior with Explicit Congestion Notification (ECN)", IEEE ICC 2002, New York, New York, USA, May 2002, <http://dx.doi.org/10.1109/ICC.2002.997262>.

[PAM2015] Trammell, B., Kuhlewind, M., Boppart, D., Learmonth, I., Fairhurst, G., and R. Scheffenegger, "Enabling Internet-wide Deployment of Explicit Congestion Notification", Proceedings of the 2015 Passive and Active Measurement Conference, New York, March 2015, <http://ecn.ethz.ch/ecn-pam15.pdf>.

[RFC7713] Mathis, M. and B. Briscoe, "Congestion Exposure (ConEx) Concepts, Abstract Mechanism, and Requirements", RFC 7713, DOI 10.17487/RFC7713, December 2015, <http://www.rfc-editor.org/info/rfc7713>.

[WWDC2015]
Lakhera, P. and S. Cheshire, "Your App and Next Generation Networks", Apple Worldwide Developers Conference 2015, San Francisco, USA, June 2015, <https://developer.apple.com/videos/wwdc/2015/?id=719>.

Authors' Addresses

Naeem Khademi
University of Oslo
PO Box 1080 Blindern
Oslo, N-0316
Norway

Email: naeemk@ifi.uio.no

Michael Welzl
University of Oslo
PO Box 1080 Blindern
Oslo, N-0316
Norway

Email: michawe@ifi.uio.no

Grenville Armitage
Centre for Advanced Internet Architectures
Swinburne University of Technology
PO Box 218
John Street, Hawthorn
Victoria, 3122
Australia

Email: garmitage@swin.edu.au

Godred Fairhurst
University of Aberdeen
School of Engineering, Fraser Noble Building
Aberdeen, AB24 3UE
UK

Email: gorry@erg.abdn.ac.uk