

Network Working Group
Internet-Draft
Updates: [3168](#) (if approved)
Intended status: Experimental
Expires: October 5, 2016

N. Khademi
M. Welzl
University of Oslo
G. Armitage
Swinburne University of Technology
G. Fairhurst
University of Aberdeen
April 03, 2016

TCP Alternative Backoff with ECN (ABE)
draft-khademi-alternativebackoff-ecn-03

Abstract

This memo provides an experimental update to [RFC3168](#). It updates the TCP sender-side reaction to a congestion notification received via Explicit Congestion Notification (ECN). The updated method reduces cwnd by a smaller amount than TCP does in reaction to loss. The intention is to achieve good throughput when the queue at the bottleneck is smaller than the bandwidth-delay-product of the connection. This is more likely when an Active Queue Management (AQM) mechanism has used ECN to CE-mark a packet, than when a packet was lost. Future versions of this document will discuss SCTP as well as other transports using ECN.

Status of This Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

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."

This Internet-Draft will expire on October 5, 2016.

Copyright Notice

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.

Table of Contents

1.	Introduction	2
2.	Discussion	3
2.1.	Why use ECN to vary the degree of backoff?	3
2.2.	Choice of ABE multiplier	4
3.	NEW: Updating the Sender-side ECN Reaction	5
3.1.	RFC 2119	6
3.2.	Update to RFC 3168	6
3.3.	Status of the Update	7
4.	Acknowledgements	7
5.	IANA Considerations	7
6.	Security Considerations	7
7.	References	8
7.1.	Normative References	8
7.2.	Informative References	8
	Authors' Addresses	9

[1.](#) Introduction

Explicit Congestion Notification (ECN) is specified in [[RFC3168](#)]. It 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 drop ECN-capable packets when incipient congestion is detected. When an ECN-capable transport is used over a path that supports ECN, it 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 $\max(\text{FlightSize} / 2, 2 * \text{SMSS})$ in response to 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 (halving cwnd and ssthresh after packet loss) is not the only available strategy. As defined in [\[ID.CUBIC\]](#), CUBIC multiplies the current cwnd by 0.8 in response to loss (although the Linux implementation of CUBIC has used a multiplier of 0.7 since kernel version 2.6.25 released in 2008). Consequently, CUBIC utilise 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 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.

Results reported in [\[ABE2015\]](#) show significant benefits (improved throughput) when reacting to ECN-Echo by multiplying cwnd and ssthresh with a value in the range [0.7..0.85]. [Section 2](#) describes the rationale for this change. [Section 3](#) specifies a change to the TCP sender backoff behaviour in response to an indication that CE-marks have been received by the receiver.

2. Discussion

Much of the background to this proposal can be found in [\[ABE2015\]](#). Using a mix of experiments, theory and simulations with standard NewReno and CUBIC, [\[ABE2015\]](#) recommends enabling ECN and "...letting individual TCP senders use a larger multiplicative decrease factor in reaction to ECN CE-marks from AQM-enabled bottlenecks." Such a change is noted to result in "...significant performance gains in lightly-multiplexed scenarios, without losing the delay-reduction benefits of deploying CoDel or PIE."

[2.1.](#) Why use ECN to vary the degree of backoff?

The classic rule-of-thumb dictates 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 enough 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 high BDP paths. Conventional TCP backoff in such cases leads to gaps in packet transmission and under-utilisation of the path.

In an ideal world, the TCP sender would adapt its backoff strategy to match the effective depth at which a bottleneck begins indicating congestion. In the practical world, [[ABE2015](#)] proposes using the existence of ECN CE-marks to infer whether a path's bottleneck is AQM-enabled (shallow queue) or classic DropTail (deep queue), and adjust backoff accordingly. This results in a change to [[RFC3168](#)], which recommended that TCP senders respond in the same way following indication of a received ECN CE-mark and a packet loss, making these equivalent signals of congestion. (The idea to change this behaviour pre-dates ABE. [[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 deal with RED-based bottlenecks that were not necessarily configured to emulate a shallow queue.)

[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. 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.

[2.2.](#) Choice of ABE multiplier

ABE decouples a TCP sender's reaction to loss and ECN CE-marks. The description respectively uses β_{loss} and β_{ecn} to refer to the multiplicative decrease factors applied in response to packet loss and in response to an indication of a received CN CE-mark on an ECN-enabled TCP connection (based on the terms used in [[ABE2015](#)]).

For non-ECN-enabled TCP connections, no ECN CE-marks are received and only β_{loss} applies.

In other words, in response to detected loss:

$$\text{FlightSize}_{(n+1)} = \text{FlightSize}_n * \beta_{\text{loss}}$$

and in response to an indication of a received ECN CE-mark:

$$\text{FlightSize}_{(n+1)} = \text{FlightSize}_n * \beta_{\text{ecn}}$$

where, as in [\[RFC5681\]](#), FlightSize is the amount of outstanding data in the network, upper-bounded by the sender's congestion window (cwnd) and the receiver's advertised window (rwnd). The higher the values of β_* , the less aggressive the response of any individual backoff event.

The appropriate choice for β_{loss} and β_{ecn} values is a balancing act between path utilisation and draining the bottleneck queue. More aggressive backoff (smaller β_*) risks underutilising the path, while less aggressive backoff (larger β_*) can result in slower draining of the bottleneck queue.

The Internet has already been running with at least two different β_{loss} values for several years: the value in [\[RFC5681\]](#) is 0.5, and Linux CUBIC uses 0.7. ABE proposes no change to β_{loss} used by any current TCP implementations.

β_{ecn} depends on how we want to optimise the response of a TCP connection to shallow AQM marking thresholds. β_{loss} reflects the preferred response of each TCP algorithm when faced with exhaustion of buffers (of unknown depth) signalled by packet loss. Consequently, for any given TCP algorithm the choice of β_{ecn} is likely to be algorithm-specific, rather than a constant multiple of the algorithm's existing β_{loss} .

A range of experiments (section IV, [\[ABE2015\]](#)) with NewReno and CUBIC over CoDel and PIE in lightly multiplexed scenarios have explored this choice of parameter. These experiments indicate that CUBIC connections benefit from β_{ecn} of 0.85 (cf. $\beta_{\text{loss}} = 0.7$), and NewReno connections see improvements with β_{ecn} in the range 0.7 to 0.85 (c.f., $\beta_{\text{loss}} = 0.5$).

3. NEW: Updating the Sender-side ECN Reaction

This section specifies an experimental update to [\[RFC3168\]](#).

3.1. 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](#)].

3.2. Update to RFC 3168

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.

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 this by replacing it with 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. This indication of congestion could be treated in the same way as a congestion loss, however reception of the ECN-Echo flag SHOULD produce a reduction in FlightSize that is less than the reduction had the flow experienced loss. The reduction needs to be sufficient to allow flows sharing a bottleneck to increase their share of the capacity. This reduction MUST be less than 0.85 (at least a 15% reduction).

An ECN-capable network device cannot eliminate the possibility of loss, because a drop may 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 explicitly does not update this behaviour.

In addition, this document RECOMMENDS that experimental deployments method multiply the FlightSize by 0.8 and reduce the slow start threshold 'ssthresh' in response to reception of a TCP segment that sets the ECN-Echo flag."

3.3. Status of the Update

This update is a sender-side only change. Like other changes to congestion-control algorithms it does not require any change to the TCP receiver or to network devices (except to enable an ECN-marking algorithm [[RFC3168](#)] [[RFC7567](#)]). If the method is only deployed by some TCP senders, and not by others, the senders that use this method can gain advantage, possibly at the expense of other flows that do not use this updated method. This advantage applies only to ECN-marked packets and not to loss indications. Hence, the new method can not lead to congestion collapse.

The present specification has been assigned an Experimental status, to provide Internet deployment experience before being proposed as a Standards-Track update.

4. Acknowledgements

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.

The authors would like to thank the following people for their contributions to [[ABE2015](#)]: Chamil Kulatunga, David Ros, Stein Gjessing, Sebastian Zander. Thanks to (in alphabetical order) Bob Briscoe, John Leslie, Dave Taht and the TCPM WG for providing valuable feedback on this document.

The authors would like to thank feedback on the congestion control behaviour specified in this update received from the IRTF Internet Congestion Control Research Group (ICCRG).

5. IANA Considerations

XX RFC ED - PLEASE REMOVE THIS SECTION XXX

This memo includes no request to IANA.

6. Security Considerations

The described method is a sender-side only transport change, and does not change the protocol messages exchanged. The security considerations of [RFC 3819](#) therefore still apply.

This document describes a change to TCP congestion control with ECN that will typically lead to a change in the capacity achieved when flows share a network bottleneck. Similar unfairness in the way that

capacity is shared is also exhibited by other congestion control mechanisms that have been in use in the Internet for many years (e.g., CUBIC [[ID.CUBIC](#)]). Unfairness may also be a result of other factors, including the round trip time experienced by a flow. This advantage applies only to ECN-marked packets and not to loss indications, and will therefore not lead to congestion collapse.

[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>>.
- [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.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.CoDel]

Nichols, K., Jacobson, V., McGregor, V., and J. Iyengar, "The Benefits of using Explicit Congestion Notification (ECN)", Internet-draft, IETF work-in-progress [draft-ietf-aqm-codel-02](#), December 2015.

[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-03](#), 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>>.

[ID.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-00](#), June 2015.

[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>>.

[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

