Internet Engineering Task Force (IETF)
Request for Comments: 8511
Category: Experimental
ISSN: 2070-1721

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

December 2018

TCP Alternative Backoff with ECN (ABE)

Abstract

Active Queue Management (AQM) mechanisms allow for burst tolerance while enforcing short queues to minimise the time that packets spend enqueued at a bottleneck. This can cause noticeable performance degradation for TCP connections traversing such a bottleneck, especially if there are only a few flows or their bandwidth-delay product (BDP) is large. The reception of a Congestion Experienced (CE) Explicit Congestion Notification (ECN) mark indicates that an AQM mechanism is used at the bottleneck, and the bottleneck network queue is therefore likely to be short. Feedback of this signal allows the TCP sender-side ECN reaction in congestion avoidance to reduce the Congestion Window (cwnd) by a smaller amount than the congestion control algorithm’s reaction to inferred packet loss. Therefore, this specification defines an experimental change to the TCP reaction specified in RFC 3168, as permitted by RFC 8311.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

Explicit Congestion Notification (ECN) [RFC3168] makes it possible for an Active Queue Management (AQM) mechanism to signal the presence of incipient congestion without necessarily incurring packet loss. This lets the network deliver some packets to an application that would have been dropped if the application or transport did not support ECN. This packet loss reduction is the most obvious benefit of ECN, but it is often relatively modest. Other benefits of deploying ECN have been documented in [RFC8087].

The rules for ECN were originally written to be very conservative, and they required the congestion control algorithms of ECN-Capable Transport (ECT) protocols to treat indications of congestion signalled by ECN exactly the same as they would treat an inferred packet loss [RFC3168]. Research has demonstrated the benefits of reducing network delays that are caused by interaction of loss-based TCP congestion control and excessive buffering [BUFFERBLOAT]. This has led to the creation of AQM mechanisms like Proportional Integral Controller Enhanced (PIE) [RFC8033] and Controlling Queue Delay (CoDel) [RFC8289], which prevent bloated queues that are common with unmanaged and excessively large buffers deployed across the Internet [BUFFERBLOAT].

The AQM mechanisms mentioned above aim to keep a sustained queue short while tolerating transient (short-term) packet bursts. However, currently used loss-based congestion control mechanisms are not always able to effectively utilise a bottleneck link where there are short queues. For example, a TCP sender using the Reno congestion control needs to be able to store at least an end-to-end bandwidth-delay product (BDP) worth of data at the bottleneck buffer if it is to maintain full path utilisation in the face of loss-induced reduction of the congestion window (cwnd) [RFC5681]. This amount of buffering effectively doubles the amount of data that can be in flight and the maximum round-trip time (RTT) experienced by the TCP sender.

Modern AQM mechanisms can use ECN to signal the early signs of impending queue buildup long before a tail-drop queue would be forced to resort to dropping packets. It is therefore appropriate for the transport protocol congestion control algorithm to have a more measured response when it receives an indication with an early warning of congestion after the remote endpoint receives an ECN CE-marked packet. Recognizing these changes in modern AQM practices, the strict requirement that ECN CE signals be treated identically to inferred packet loss has been relaxed [RFC8311]. This document therefore defines a new sender-side-only congestion control response called "ABE" (Alternative Backoff with ECN). ABE improves TCP’s average throughput when routers use AQM-controlled buffers that allow only for short queues.

2. Definitions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Specification

This specification changes the congestion control algorithm of an ECN-Capable TCP transport protocol by changing the TCP-sender response to feedback from the TCP receiver that indicates the reception of a CE-marked packet, i.e., receipt of a packet with the ECN-Echo flag (defined in [RFC3168]) set, following the process defined in [RFC8311].

The TCP-sender response is currently specified in Section 6.1.2 of the ECN specification [RFC3168] and has been slightly updated by Section 4.1 of [RFC8311] to read as:

> 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", unless otherwise specified by an Experimental RFC in the IETF document stream.

As permitted by RFC 8311, this document specifies a sender-side change to TCP where receipt of a packet with the ECN-Echo flag SHOULD trigger the TCP source to set the slow start threshold (ssthresh) to 0.8 times the FlightSize, with a lower bound of 2 * SMSS applied to the result (where SMSS stands for Sender Maximum Segment Size)). As in [RFC5681], the TCP sender also reduces the cwnd value to no more than the new ssthresh value. Section 6.1.2 of RFC 3168 provides guidance on setting a cwnd less than 2 * SMSS.

3.1. Choice of ABE Multiplier

ABE decouples the reaction of a TCP sender to inferred packet loss from the indication of ECN-signalled congestion in the congestion avoidance phase. To achieve this, ABE uses a different scaling factor for Equation 4 in Section 3.1 of [RFC5681]. The description respectively uses beta_{loss} and beta_{ecn} to refer to the multiplicative decrease factors applied in response to inferred packet loss, and in response to a receiver indicating ECN-signalled congestion. For non-ECN-enabled TCP connections, only beta_{loss} applies.

In other words, in response to inferred packet loss:

ssthresh = max (FlightSize * beta_{loss}, 2 * SMSS)

and in response to an indication of an ECN-signalled congestion:

ssthresh = max (FlightSize * beta_{ecn}, 2 * SMSS)

and

cwnd = ssthresh

(If ssthresh == 2 * SMSS, Section 6.1.2 of RFC 3168 provides guidance on setting a cwnd lower than 2 * SMSS.)

where FlightSize is the amount of outstanding data in the network, upper-bounded by the smaller of the sender's cwnd and the receiver's advertised window (rwnd) [RFC5681]. The higher the values of beta_{loss} and beta_{ecn}, the less aggressive the response of any individual backoff event.

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

The Internet has already been running with at least two different beta_{loss} values for several years: the standard value is 0.5 [RFC5681], and the Linux implementation of CUBIC [RFC8312] has used a multiplier of 0.7 since kernel version 2.6.25 released in 2008. ABE does not change the value of beta_{loss} used by current TCP implementations.

The recommendation in this document specifies a value of beta_{ecn}=0.8. This recommended beta_{ecn} value is only applicable for the standard TCP congestion control [RFC5681]. The selection of beta_{ecn} enables tuning the response of a TCP connection to shallow AQM-marking thresholds. beta_{loss} characterizes the response of a congestion control algorithm to packet loss, i.e., exhaustion of buffers (of unknown depth). Different values for beta_{loss} have been suggested for TCP congestion control algorithms. Consequently, beta_{ecn} is likely to be an algorithm-specific parameter rather than a constant multiple of the algorithm's existing beta_{loss}.

A range of tests (Section IV of [ABE2017]) with NewReno and CUBIC over CoDel and PIE in lightly multiplexed scenarios have explored this choice of parameter. The results of these tests indicate that CUBIC connections benefit from beta_{ecn} of 0.85 (cf. beta_{loss} = 0.7), and NewReno connections see improvements with beta_{ecn} in the range 0.7 to 0.85 (cf. beta_{loss} = 0.5).

4. Discussion

Much of the technical background for ABE can be found in [ABE2017], which uses a mix of experiments, theory, and simulations with NewReno [RFC5681] and CUBIC [RFC8312] to evaluate its performance. ABE was shown to present significant performance gains in lightly-multiplexed (few concurrent flows) scenarios, without losing the delay-reduction benefits of deploying CoDel or PIE. The performance improvement is achieved when reacting to ECN-Echo in congestion avoidance (when ssthresh > cwnd) by multiplying cwnd and ssthresh with a value in the range [0.7,0.85]. Applying ABE when cwnd is smaller than or equal to ssthresh is not currently recommended, but its use in that scenario may benefit from additional attention, experimentation, and specification.

4.1. Rationale for Using ECN to Vary the Degree of Backoff

AQM mechanisms such as CoDel [RFC8289] and PIE [RFC8033] set a delay target in routers and 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 bottleneck with a short queue (Section II of [ABE2017]) while also allowing short traffic bursts into the queue. This provides acceptable performance for TCP connections over a path with a low BDP, or in highly multiplexed scenarios (many concurrent transport flows). However, in a lightly multiplexed case over a path with a large BDP, conventional TCP backoff leads to gaps in packet transmission and underutilisation of the path.

Instead of discarding packets, an AQM mechanism is allowed to mark ECN-Capable packets with an ECN CE mark. The reception of CE-mark feedback not only indicates congestion on the network path, it also indicates that an AQM mechanism exists at the bottleneck along the path. Therefore, the CE mark likely came from a bottleneck with a controlled short queue. Reacting differently to an ECN-signalled congestion than to an inferred packet loss can then yield the benefit of a reduced backoff when queues are short. Using ECN can also be advantageous for several other reasons [RFC8087].

The idea of reacting differently to inferred packet loss and detection of an ECN-signalled congestion predates this specification, e.g., previous research proposed using ECN CE-marked feedback to modify TCP congestion control behaviour via a larger multiplicative decrease factor in conjunction with a smaller additive increase factor [ICC2002]. The goal of this former work was to operate across AQM bottlenecks (using Random Early Detection (RED)) that were not necessarily configured to emulate a short queue. (The current usage of RED as an Internet AQM method is limited [RFC7567].)

4.2. An RTT-Based Response to Indicated Congestion

This specification applies to the use of ECN feedback as defined in [RFC3168], which specifies a response to indicated congestion that is no more frequent than once per path round-trip time. Since ABE responds to indicated congestion once per RTT, it does not respond to any further loss within the same RTT because an ABE sender has already reduced the congestion window. If congestion persists after such reduction, ABE continues to reduce the congestion window in each consecutive RTT. This consecutive reduction can protect the network against long-standing unfairness in the case of AQM algorithms that do not keep a small average queue length. The mechanism does not rely on Accurate ECN [ACC-ECN-FEEDBACK].

In contrast, transport protocol mechanisms can also be designed to utilise more frequent and detailed ECN feedback (e.g., Accurate ECN [ACC-ECN-FEEDBACK]), which then permit a congestion control response that adjusts the sending rate more frequently. Data Center TCP (DCTCP) [RFC8257] is an example of this approach.

5. ABE Deployment Requirements

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. It does not require any ABE-specific changes in routers or the use of Accurate ECN feedback [ACC-ECN-FEEDBACK] by a receiver.

If the method is only deployed by some senders, and not by others, the senders using it can gain some advantage, possibly at the expense of other flows that do not use this updated method. Because this advantage applies only to ECN-marked packets and not to packet-loss indications, an ECN-Capable bottleneck will still fall back to dropping packets if a TCP sender using ABE is too aggressive. The result is no different than if the TCP sender were using traditional loss-based congestion control.

When used with bottlenecks that do not support ECN marking, the specification does not modify the transport protocol.

6. ABE Experiment Goals

[RFC3168] states that the congestion control response following an indication of ECN-signalled congestion is the same as the response to a dropped packet. [RFC8311] updates this specification to allow systems to provide a different behaviour when they experience ECN-signalled congestion rather than packet loss. The present specification defines such an experiment and is an Experimental RFC. We expect to propose it as a Standards-Track document in the future.

The purpose of the Internet experiment is to collect experience with the deployment of ABE and confirm acceptable safety in deployed networks that use this update to TCP congestion control. To evaluate ABE, this experiment requires support in AQM routers for the ECN-marking of packets carrying the ECN-Capable Transport codepoint ECT(0) [RFC3168].

The result of this Internet experiment ought to include an investigation of the implications of experiencing an ECN-CE mark followed by loss within the same RTT. At the end of the experiment, this will be reported to the TCPM Working Group or the IESG.

ABE is implemented as a patch for Linux and FreeBSD. This is meant for research and experimentation and is available for download at <https://heim.ifi.uio.no/michawe/research/abe/>. This code was used to produce the test results that are reported in [ABE2017]. The FreeBSD code was committed to the mainline kernel on March 19, 2018 [ABE-REVISION].

7. IANA Considerations

This document has no IANA actions.

8. Security Considerations

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

This is 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. This could result in some flows receiving more than their fair share of capacity. 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 [RFC8312]). Unfairness may also be a result of other factors, including the round-trip time experienced by a flow. ABE applies only when ECN-marked packets are received, not when packets are lost. Therefore, use of ABE cannot lead to congestion collapse.

9. References

9.1. 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, <https://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, <https://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, <https://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, <https://www.rfc-editor.org/info/rfc7567>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8257] Bensley, S., Thaler, D., Balasubramanian, P., Eggert, L., and G. Judd, "Data Center TCP (DCTCP): TCP Congestion Control for Data Centers", RFC 8257, DOI 10.17487/RFC8257, October 2017, <https://www.rfc-editor.org/info/rfc8257>.

[RFC8311] Black, D., "Relaxing Restrictions on Explicit Congestion Notification (ECN) Experimentation", RFC 8311, DOI 10.17487/RFC8311, January 2018, <https://www.rfc-editor.org/info/rfc8311>.

9.2. Informative References

[ABE-REVISION] Stewart, L., "ABE patch review in FreeBSD", Revision 331214, March 2018, <https://svnweb.freebsd.org/base?view=revision&revision=331214>.

[ABE2017] Khademi, N., Armitage, G., Welzl, M., Zander, S., Fairhurst, G., and D. Ros, "Alternative backoff: Achieving low latency and high throughput with ECN and AQM", IFIP Networking Conference and Workshops Stockholm, Sweden, DOI 10.23919/IFIPNetworking.2017.8264863, June 2017.

[ACC-ECN-FEEDBACK] Briscoe, B., Kuehlewind, M., and R. Scheffenegger, "More Accurate ECN Feedback in TCP", Work in Progress, draft-ietf-tcpm-accurate-ecn-07, July 2018.

[BUFFERBLOAT] Gettys, J. and K. Nichols, "Bufferbloat: Dark Buffers in the Internet", ACM Queue, Volume 9, Issue 11, DOI 10.1145/2063166.2071893, November 2011, <https://queue.acm.org/detail.cfm?id=2071893>.

[ICC2002] Kwon, M. and S. Fahmy, "TCP increase/decrease behavior with explicit congestion notification (ECN)", 2002 IEEE International Conference on Communications Conference Proceedings, ICC 2002, Cat. No.02CH37333, DOI 10.1109/ICC.2002.997262, May 2002, <http://dx.doi.org/10.1109/ICC.2002.997262>.

[RFC8033] Pan, R., Natarajan, P., Baker, F., and G. White, "Proportional Integral Controller Enhanced (PIE): A Lightweight Control Scheme to Address the Bufferbloat Problem", RFC 8033, DOI 10.17487/RFC8033, February 2017, <https://www.rfc-editor.org/info/rfc8033>.

[RFC8087] Fairhurst, G. and M. Welzl, "The Benefits of Using Explicit Congestion Notification (ECN)", RFC 8087, DOI 10.17487/RFC8087, March 2017, <https://www.rfc-editor.org/info/rfc8087>.

[RFC8289] Nichols, K., Jacobson, V., McGregor, A., Ed., and J. Iyengar, Ed., "Controlled Delay Active Queue Management", RFC 8289, DOI 10.17487/RFC8289, January 2018, <https://www.rfc-editor.org/info/rfc8289>.

[RFC8312] Rhee, I., Xu, L., Ha, S., Zimmermann, A., Eggert, L., and R. Scheffenegger, "CUBIC for Fast Long-Distance Networks", RFC 8312, DOI 10.17487/RFC8312, February 2018, <https://www.rfc-editor.org/info/rfc8312>.

Acknowledgements

Authors N. Khademi, M. Welzl, and G. Fairhurst were partly 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.

Author G. Armitage performed most of his work on this document while employed by Swinburne University of Technology, Melbourne, Australia.

The authors would like to thank Stuart Cheshire for many suggestions when revising this document. They would also like to thank the following people for their contributions to [ABE2017]: Chamil Kulatunga, David Ros, Stein Gjessing, and Sebastian Zander. Thanks also to (in alphabetical order) David Black, Roland Bless, Bob Briscoe, Markku Kojo, John Leslie, Lawrence Stewart, and the TCPM Working Group for providing valuable feedback on this document.

Finally, the authors would like to thank everyone who provided feedback on the congestion control behaviour specified in this document that was received from the IRTF Internet Congestion Control Research Group (ICCRG).

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
Netflix Inc.

Email: garmitage@netflix.com

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

Email: gorry@erg.abdn.ac.uk