Network Working Group N. Khademi
Internet-Draft M. Welzl
Intended status: Experimental University of Oslo
Expires: September 21, 2018 G. Armitage
Swinburne University of Technology
G. Fairhurst
University of Aberdeen
March 20, 2018

TCP Alternative Backoff with ECN (ABE)
draft-ietf-tcpm-alternativebackoff-ecn-07

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 is large. An Explicit Congestion Notification (ECN) signal indicates that an AQM mechanism is used at the bottleneck, and therefore the bottleneck network queue is likely to be short. This document therefore proposes an update to RFC3168, which changes 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.

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 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 [RFC8087].

The rules for ECN were originally written to be very conservative, and required the congestion control algorithms of ECN-Capable

transport protocols to treat ECN congestion signals 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 new AQM mechanisms like PIE [RFC8033] and CoDel [CODEL2012][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 cannot always utilise a bottleneck link well where there are short queues. For example, a TCP sender must 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 cwnd [RFC5681], which effectively doubles the amount of data that can be in flight, the maximum round-trip time (RTT) experience, and the path's effective RTT using the network path.

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 an early-warning signal of congestion is received in the form of an ECN CE-marked packet. Recognizing these changes in modern AQM practices, more recent rules have relaxed the strict requirement that ECN signals be treated identically to inferred packet loss [RFC8311]. Following these newer, more flexible rules, this document 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 for short queues only.

2. Definitions

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].

3. Specification

This specification updates 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 reception

of a CE-marked packet, i.e., receipt of a packet with the ECN-Echo flag (defined in [RFC3168]) set.

It updates the following text in section 6.1.2 of the ECN specification [RFC3168] :

> 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".

Replacing this with:

> 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. As in [RFC5681], the TCP sender also reduces the cwnd value to no more than the new ssthresh value. RFC 3168 section 6.1.2 provides guidance on setting a cwnd less than 2 * SMSS.

4. Discussion

Much of the technical background to ABE can be found in a research paper [ABE2017]. This paper used a mix of experiments, theory and simulations with NewReno [RFC5681] and CUBIC [RFC8312] to evaluate the technique. The technique 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 <= ssthresh is not currently recommended, but may benefit from additional attention, experimentation and specification.

4.1. Why Use 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, [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 under-utilisation 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 a 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, and hence 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 back-off 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 pre-dates this document. For example, 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. Focus on ECN as Defined in RFC3168

Some transport protocol mechanisms rely on ECN semantics that differ from the original ECN definition [RFC3168]. For instance, Accurate ECN [I-D.ietf-tcpm-accurate-ecn] permits more frequent and detailed feedback. Use of such mechanisms (including Accurate ECN, Datacenter TCP (DCTCP) [RFC8257], or Congestion Exposure (ConEx) [RFC7713]) is out of scope for this document. This specification focuses on ECN as defined in [RFC3168].

4.3. Choice of ABE Multiplier

ABE decouples the reaction of a TCP sender to inferred packet loss and ECN-signalled congestion in the congestion avoidance phase. To achieve this, ABE uses a different scaling factor in 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, RFC 3168 section 6.1.2 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 underutilising 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 proposes no change to beta_{loss} used by current TCP implementations.

The recommendation in Section 3 in this document corresponds to 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, [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).

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 [I-D.ietf-tcpm-accurate-ecn] by a receiver.

RFC3168 states that the congestion control response to an ECN-signalled congestion is the same as the response to a dropped packet [RFC3168]. [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 has thus been assigned an Experimental status before being proposed as a Standards-Track update.

The purpose of the Internet experiment is to collect experience with deployment of ABE, and confirm the safety in deployed networks using this update to TCP congestion control.

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

To evaluate the benefit, this experiment therefore requires support in AQM routers for ECN-marking of packets carrying the ECN-Capable Transport, ECT(0), codepoint [RFC3168].

If the method is only deployed by some senders, and not by others, the senders that use this method 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 an TCP sender using ABE is too aggressive, and the result is no different than if the TCP sender was using traditional loss-based congestion control.

A TCP sender reacts to loss or ECN marks only once per round-trip time. Hence, if a sender would first be notified of an ECN mark and then learn about loss in the same round-trip, it would only react to the first notification (ECN) but not to the second (loss). RFC3168 specified a reaction to ECN that was equal to the reaction to loss [RFC3168].

ABE also responds to congestion once per RTT, and therefore it does not respond to further loss within the same RTT, since ABE 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 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 WG (or IESG).

6. 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 Stuart Cheshire for many suggestions when revising the draft, and the following people for their contributions to [ABE2017]: Chamil Kulatunga, David Ros, Stein Gjessing, Sebastian Zander. Thanks also to (in alphabetical order) Roland Bless, Bob Briscoe, David Black, Markku Kojo, John Leslie, Lawrence Stewart, Dave Taht and the TCPM working group for providing valuable feedback on this document.

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

7. IANA Considerations

XX RFC ED - PLEASE REMOVE THIS SECTION XXX

This document includes no request to IANA.

8. Implementation Status

ABE is implemented as a patch for Linux and FreeBSD. It is meant for research and available for download from http://heim.ifi.uio.no/naeemk/research/ABE/. This code was used to produce the test results that are reported in [ABE2017]. The FreeBSD code has been committed to the mainline kernel on March 19, 2018 [ABE-FreeBSD].

9. Security Considerations

The described method is a sender-side only transport change, and does not change the protocol messages exchanged. The security considerations for ECN [RFC3168] therefore 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, hence use of ABE cannot lead to congestion collapse.

10. Revision Information

XX RFC ED - PLEASE REMOVE THIS SECTION XXX

-07. Addressed comments following WGLC.

o Updated Reference citations

o Removed paragraph containing a wrong statement related to timeout in section 4.1.

o Discuss what happens when cwnd <= ssthresh

o Added text on Concern about lower bound of 2*SMSS

-06. Addressed Michael Scharf's comments.

-05. Refined the description of the experiment based on feedback at IETF-100. Incorporated comments from David Black.

-04. Incorporates review comments from Lawrence Stewart and the remaining comments from Roland Bless. References are updated.

-03. Several review comments from Roland Bless are addressed. Consistent terminology and equations. Clarification on the scope of recommended beta_{ecn} value.

-02. Corrected the equations in Section 4.3. Updated the affiliations. Lower bound for cwnd is defined. A recommendation for window-based transport protocols is changed to cover all transport protocols that implement a congestion control reduction to an ECN

congestion signal. Added text about ABE's FreeBSD mainline kernel status including a reference to the FreeBSD code review page. References are updated.

-01. Text improved, mainly incorporating comments from Stuart Cheshire. The reference to a technical report has been updated to a published version of the tests [ABE2017]. Used "AQM Mechanism" throughout in place of other alternatives, and more consistent use of technical language and clarification on the intended purpose of the experiments required by EXP status. There was no change to the technical content.

-00. draft-ietf-tcpm-alternativebackoff-ecn-00 replaces draft-khademi-tcpm-alternativebackoff-ecn-01. Text describing the nature of the experiment was added.

Individual draft -01. This I-D now refers to draft-black-tsvwg-ecn-experimentation-02, which replaces draft-khademi-tsvwg-ecn-response-00 to make a broader update to RFC3168 for the sake of allowing experiments. As a result, some of the motivating and discussing text that was moved from draft-khademi-alternativebackoff-ecn-03 to draft-khademi-tsvwg-ecn-response-00 has now been re-inserted here.

Individual draft -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.

11. References

11.1. 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>.

[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>.

11.2. Informative References

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

[ABE2017] Khademi, N., Armitage, G., Welzl, M., Fairhurst, G., Zander, S., and D. Ros, "Alternative Backoff: Achieving Low Latency and High Throughput with ECN and AQM", IFIP NETWORKING 2017, Stockholm, Sweden, June 2017.

[BUFFERBLOAT] Gettys, J. and K. Nichols, "Bufferbloat: Dark Buffers in the Internet", November 2011.

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

[I-D.ietf-tcpm-accurate-ecn] Briscoe, B., Kuehlewind, M., and R. Scheffenegger, "More Accurate ECN Feedback in TCP", draft-ietf-tcpm-accurate-ecn-06 (work in progress), March 2018.

[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>.

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

[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>.

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
Internet For Things (I4T) Research Group
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