TCP Maintenance and Minor Extensions
Internet-Draft
Intended status: Experimental
Expires: 7 May 2020

R.W. Grimes
P. Heist

4 November 2019

Some Congestion Experienced in TCP
draft-grimes-tcpm-tcpsce-01

Abstract

This memo classifies a TCP code point ESCE ("Echo Some Congestion Experienced") for use in feedback of IP code point SCE ("Some Congestion Experienced").

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 https://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 7 May 2020.

Table of Contents

1. Terminology

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 [RFC2119] and [RFC8174] when, and only when, they appear in all capitals, as shown here.

2. Introduction

This memo requests a TCP header codepoint for use as ESCE.

This memo limits its scope to the definition of the TCP codepoint ESCE, with a few brief illustrations of how it may be used.

SCE provides early and proportional feedback to the CC (congestion control) algorithms for transport protocols, including but not limited to TCP. The [sce-repo] is a Linux kernel modified to support SCE, including:

* Enhancements to Linux's [cake] (Common Applications Kept Enhanced) AQM to support SCE signaling

* Modifications to the TCP receive path to reflect SCE signals back to the sender

* The addition of three new TCP CC algorithms that modify the originals to add SCE support: Reno-SCE, DCTCP-SCE and Cubic-SCE (work in progress as of this writing)

3. Background

[I-D.morton-tsvwg-sce] defines the IP SCE codepoint.

4. TCP Receiver

The mechanism defined to feed back SCE signals to the sender explicitly makes use of the ESCE ("Echo Some Congestion Experienced") code point in the TCP header.

4.1. Single ACK implementation

Upon receipt of a packet an ACK is immediatly generated, the SCE codepoint is copied into the ESCE codepoint of the ACK. This keeps the count of bytes SCE marked or not marked properly reflected in the ACK packet(s). This valid implementation has the downside of increasing ACK traffic. This implementation is NOT RECOMMENDED, but useful for experimental work.

4.2. Simple Delayed ACK implementation

Upon receipt of a packet without an SCE codepoint traditional delayed ACK processing is performed. Upon receipt of a packet with an SCE codepoint immediate ACK processing SHOULD be done, this allows some delaying of ACK's, but creates earlier feedback of the congested state. This has the negative effect of over signalling ESCE.

4.3. Dithered Delayed ACK implementation

Upon receipt of a packet the SCE codepoint is stored in the TCP state. Multiple packets state may be stored. Upon generation of an ACK, normal or delayed, the stored SCE state is used to set the state of ESCE. If no SCE state is in the TCP state, then the ESCE code point MUST NOT be set. If all of the packets to be ACKed have SCE state set then the ESCE code point MUST be set in the ACK. If some of the packets to be ACKed have SCE state set then some proportional number of ACK packets SHOULD be sent with the ESCE code point set. Though this may defer a ESCE congestion signal when there is not a next packet for some time it is generally accepted that such sparse flows are not the source of congestion and thus the delayed signal is of low impact. The goal is to have the same number of bytes marked with ESCE as arrived with SCE.

4.4. Advanced ACK implementation

The Advanced ACK implementation actually immediately flushes any pending ACK's up to the _previous_ segment when the state of the SCE marking _changes_, allowing consecutive packets with the same SCE state to be coalesced by the normal delayed-ack logic. The ACK volume is then inflated only slightly compared to an unmarked connection, and may actually involve fewer acks than a connection involving CE marks or losses, during which delayed acks are temporarily disabled.

4.5. ACK Thinning

Ack thinning is something that has been considered, given that [cake] includes an optional ack-filter which does thinning. We have, for example, added consideration of the ESCE bit to Cake's ack-filter. Mathematically, the most extreme errors possible in either direction, due to ack thinning, are easily corrected during subsequent RTTs.

5. TCP Sender

The recommended response to each single segment marked with ESCE is to reduce cwnd by an amortised 1/sqrt(cwnd) segments. If the growth rate is greater than that provided by the Reno-linear algorithm - eg. slow-start exponential or CUBIC polynomial - then the growth rate SHOULD also be reduced.

Other responses, such as the 1/cwnd from DCTCP, are also acceptable but may perform less well.

There are no changes to the response functions with respect to CE or packet loss specificed by this draft, hence [RFC3168] and [RFC8511] are still applicable

This is still an area of continued investigation.

6. Related Work

6.1. More Accurate ECN Feedback in TCP [I-D.ietf-tcpm-accurate-ecn]

AccECN replaces the [RFC3168] definition of the ECE and CWR bits (and the former NS bit) with its own three-bit field. This new interpretation is predicated on successfully negotiating AccECN, and is not useful to SCE implementations because it provides no information about any ECT(1) codepoints received, and SCE does not need or use the extra information about CE marks that the three-bit field does provide. Hence SCE may be considered mutually exclusive with AccECN on any given connection.

AccECN supports a fallback to [RFC3168] style signalling during the three-way handshake by recognising the normal requests and responses of an [RFC3168] endpoint. SCE endpoints also exhibit [RFC3168] behaviour during the handshake, so this mutual exclusivity occurs naturally. There will therefore be no confusion on the wire between the two experiments, even though SCE does not explicitly negotiate its upgrade from plain [RFC3168] behaviour.

The latter is consistent with the (now historic) Nonce Sum specification, which also did not explicitly negotiate support, and used the same additional ECN codepoint and TCP header bit that SCE is now requesting.

7. IANA Considerations

This document requests one of the reserved bits in the TCP header, with the former TCP NS ("Nonce Sum") bit (bit 7) being suggested due to similarities with its previous usage. [RFC8311] (section 3) obsoletes the NS codepoint making it avaliable for use.

8. Security Considerations

There are no Security considerations.

9. Acknowledgements

TBD

10. Normative References

[I-D.morton-tsvwg-sce]
Morton, J. and R. Grimes, "The Some Congestion Experienced ECN Codepoint", draft-morton-tsvwg-sce-00 (work in progress), 2 July 2019, <https://www.ietf.org/archive/id/draft-morton-tsvwg-sce-00>.

[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>.

[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>.

[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. Informative References

[cake] "Cake - Common Applications Kept Enhanced", November 2019,
<http://www.bufferbloat.net/projects/codel/wiki/Cake>.

[I-D.ietf-tcpm-accurate-ecn]
Briscoe, B., Kuehlewind, M., and R. Scheffenegger, "More Accurate ECN Feedback in TCP", draft-ietf-tcpm-accurate-ecn-09 (work in progress), 8 July 2019,
<https://www.ietf.org/archive/id/draft-ietf-tcpm-accurate-ecn-09>.

[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>.

[RFC8511] Khademi, N., Welzl, M., Armitage, G., and G. Fairhurst, "TCP Alternative Backoff with ECN (ABE)", RFC 8511,
DOI 10.17487/RFC8511, December 2018,
<https://www.rfc-editor.org/info/rfc8511>.

[sce-repo] "Some Congestion Experienced Reference Implementation GitHub Repository", November 2019,
<https://github.com/chromi/sce/>.

Authors' Addresses

Rodney W. Grimes
Redacted
Portland, OR 97217
United States

Email: rgrimes@freebsd.org

Peter G. Heist
Redacted
463 11 Liberec 30
Czech Republic

Email: pete@heistp.net