Network Working Group
Internet-Draft
Intended status: Informational
Expires: January 9, 2017

M. Bagnulo
UC3M
B. Briscoe
Simula Research Lab
July 8, 2016

Adding Explicit Congestion Notification (ECN) to TCP control packets draft-bagnulo-tsvwg-generalized-ecn-01

Abstract

This documents explores the possibility of adding ECN support to TCP control packets.

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 January 9, 2017.

Table of Contents

1. Introduction

RFC3168 [RFC3168] specifies the support of Explicit Congestion Notification (ECN) to IP. By using the ECN capability, switches performing Active Queue Management (AQM) can use ECN marks instead of packets drops to signal congestion to the endpoints of a communication. This results in lower packet loss and increased performance. However, RFC3168 specifies the support of ECN in TCP data packets, but precludes the use of ECN in TCP control packets (TCP SYN, TCP SYN/ACK, pure ACKs, Window probes) and in retransmitted packets. RFC 5562 [RFC5562] is an experimental extension to ECN that enables the ECN support for TCP SYN/ACK packets.

The inability of using ECN in TCP control packets has a potential harmful effect, especially in environments where ECN support is pervasive. For example, [judd-nsdi] shows that in a data center environment where DCTCP is used (in conjunction with ECN), the the probability of being able to establish a new connection using a non-ECT-marked SYN packet drops to close to 0 when there are 16 ongoing TCP flows transmitting at full speed. In this particular context of a datacenter using DCTCP, the issue is that the proposed AQM aggressively marks packets to keep the buffer queues small and this implies that non-ECT-marked packets are in turn dropped aggressively as well, rendering nearly impossible to establish new connection when there is ongoing traffic.

These limitations are not limited to the data center environment. In any ECN deployment, non ECT marked packets suffer a penalty when they traverse a congested bottleneck. For instance, with a drop probability of 1%, 1% of connection attempts suffer a timeout before the SYN is retransmitted, which is very deterimental to the performance of short flows. Dropping TCP control traffic, such as TCP SYNs and pure ACKs have a negative effect on the overall performance of the communication, so it is beneficial to avoid it.

Finally, there are ongoing efforts to promote the adoption of DCTCP (and similar transports) over the Internet to achieve low latency for all communications [I-D.briscoe-tsvwg-aqm-tcpm-rmcat-l4s-problem]. In such approach, ECN capable packets are treated more favorably, as they are likely to experience less delay and lower packet drop probability. Preventing TCP control packets, which are critical for TCP performance, to obtain the benefits of ECN would result in degraded performance.

However, RFC3168 does not prevents from using ECN in TCP control packets lightly. It provides a number of specific reasons for each packet type. In this note, we revisit each of the arguments provided by RFC3168 and explore possibilities to enable the ECN capability in the different packet types. We do so in the context of a data center network and in the context of the public Internet.

2. The reliability argument

While for each type of packet RFC 3168 provides a set of specific arguments for preventing their marking, RFC3168 presents the reliable delivery of the congestion signal as an overarching argument that needs to be consider when trying to enable the ECT marking of TCP control packets. In particular, Section 5.2 of RFC3168 states:

> To ensure the reliable delivery of the congestion indication of the CE codepoint, an ECT codepoint MUST NOT be set in a packet unless the loss of that packet in the network would be detected by the end nodes and interpreted as an indication of congestion.

We believe this argument is overly conservative. The overall principle that should determine the level of reliability required for ECN capable packets should be the one of "do not harm". Reliable delivery of the CE codepoint is indeed paramount but the level of reliability required should be the one of the original congestion signal (i.e. the detection of the loss of the original packet). In other words, the situation without ECN is that when a packet is to be transmitted through a congested link, the packet may be dropped and that is the congestion signal sent to the endpoint. When ECN is introduced, the reliability of the delivery of the congestion signal should be no worse than without ECN. In particular, setting the CE codepoint in the very same packet seem to fulfill this criteria, since either the packet is delivered and the CE codepoint signal is delivered to the endpoint, or the packet is dropped, so the original congestion signal through the packet loss is delivered to the endpoint. Requiring more than this implies that the ECN congestion signal is delivered more reliably than the current situation, which is not a bad thing per se, but, as we describe in this memo, it

results in performance penalties that should be reconsidered in the view of current deployments.

In addition, the reliability of the delivery of the congestion signal is used an argument for not setting the ECT codepoint in TCP control packets, which effectively reduced the reliability of the transmission of these TCP control packets. There is the then a tradeoff between the reliability of the delivery of the congestion signal and the reliability of the delivery of TCP control packets. As currently specified, ECN adoption implies an increased reliability of the ECN congestion signal and a decrease in the reliability in the TCP control packets. We believe that it is possible and desirable to restore the tradeoff existent in non ECN capable networks in terms of reliability, where the congestion signal delivery is as reliable as in a non ECN capable network and so it is the delivery of TCP control packets.

3. TCP SYNs

We next describe he arguments exhibited by current specification for precluding the ECT marking of SYN packets.

In addition to the reliability argument above, RFC 5562 presents two arguments against ECT marking of SYN packets (cited verbatim):

> There are several reasons why an ECN-Capable codepoint must not be set in the IP header of the initiating TCP SYN packet. First, when the TCP SYN packet is sent, there are no guarantees that the other TCP endpoint (node B in Figure 2) is ECN-Capable, or that it would be able to understand and react if the ECN CE codepoint was set by a congested router.
>
> Second, the ECN-Capable codepoint in TCP SYN packets could be misused by malicious clients to "improve" the well-known TCP SYN attack. By setting an ECN-Capable codepoint in TCP SYN packets, a malicious host might be able to inject a large number of TCP SYN packets through a potentially congested ECN-enabled router, congesting it even further.

We next go through all the arguments stated above to enable ECT marking of SYN packets.

Argument 1: Unknown ECN capability capability at the responder. The initiator does not know whether the responder supports ECN and in particular, the initiator does not know if the responder supports ECT marked SYNs.

In the DC context, this argument does not hold (at least in single tenant DCs, possibly in multi-tenant DCs, if we assume that each tenant mostly communicates with its own VMs). The DC is a much more controlled environment than the public Internet, so the server's support of ECN can be guaranteed administratively i.e. the manager of the DC makes sure that the servers support ECN and in particular ECT marked SYN packets.

In the public Internet context, it cannot be assumed that all servers support ECN, and much less that they support ECT marked SYN packets. When sending an ECT marked SYN to a legacy responder (i.e. a responder that does not support ECT marked SYNs), different behaviours are possible.

The responder may drop the SYN (either silently or by sending a RST) or may reply with a non ECT marked SYN/ACK. If it is the latter, then this is a non-issue (the second issue presented next still applies though). If it is the former, then the initiator will have to retransmit the SYN (without the ECT mark). Depending how extended is this behaviour, this can reduce significantly the benefits of adding ECT capability to the SYN or even be detrimental for the performance. According to [ecn-pam], out of the top 1M Alexa web sites, 0,82% of IPv4 sites and 0,61% of IPv6 sites fail to establish a connection when they receive a TCP SYN with any ECN codepoint set.

If based on this data, we conclude that the fraction of fraction of servers that discard the ECT marked SYN is a non negligible, further options depend on whether they silently discard it or they send a RST back. If they send a RST back, the initiator can then send a non ECT marked SYN. In this case the penalty would be an extra RTT, which may or may not be acceptable, depending on the fraction of servers that behaves like this. If the server silently discard the ECT marked SYN, then the initiator needs to wait for the retransmission timer to expire and retransmit a non-ECT marked SYN. This is a high penalty. If this is the case, one option, would be to first send an ECT marked SYN and then a non-ECT marked SYN (possibly with a small delay between them) and establish the ECT capable connection if the former is replied. But it is questionable whether the level of failure of ECT on SYNs warrants this, particularly given failures could reduce if ECN on SYNs is standardized.

Argument 2: Loss of congestion notification in the SYN packet due to lack of support from the responder. If the ECT marked SYN packet is tagged as CE by a router along the path and the server does not support ECT marked SYN packets, even if the server replies with a SYN/ACK, the congestion information would be lost.

The accurate ECN (AccECN) proposal [I-D.ietf-tcpm-accurate-ecn] suggests a two-pringed solutions to this problem. First AccECN provides a way for the responder to feedback whether there was CE on the SYN, and second AccECN introduces a different combination of TCP header flags on the SYN/ACK so that the initiator knows whether or not the responder supports AccECN. Then if the responder does indicate that it supports AccECN the initiator can be sure that, if there is no CE feedback on the SYNACK, then there really was no CE on the SYN.

If the responder's SYN/ACK shows that it does not support AccECN, the initiator can take a conservative approach and assume the SYN was marked with CE and reduce its initial window. However, the initiator knows that congestion is not serious, because both the SYN and the SYN/ACK were delivered through the network. Therefore congestion is not serious enough for a router to have had to turn off ECN. Therefore, even a conservative initiator would not have to reduce its initial window as much as it would in response to a timeout following no response to its SYN.

Nonetheless, even a slight conservative reduction in initial window might be a significant penalty, especially in the early days of deployment, when little support for ECT SYN packets will be available. This could be mitigated by caching previous experience of which servers support AccECN.

Argument 3: DoS attacks. There are two possible DoS attacks involved in the text contained in RFC3168. On one hand, the mention about improving the well-known TCP SYN attack. The reference to the TCP SYN attack we interpret it as a reference to the TCP SYN flood attack (see https://en.wikipedia.org/wiki/SYN_flood). This attack is addressed to the responder endpoint of the connection. The argument is basically, because SYN can be used to launch attacks, their transmission should not be more reliable. While it is true that SYNs can be used to launch attacks, it is also true that SYNs are fundamental for legitimate communications, so the argument for increasing reliability of legitimate communications should take precedence. On the other hand in the RFC3168 refers about ECN capable SYN packets to congest further a bottleneck. It is not clear why a TCP SYN packet is worse than any other packet in this respect. In any case, section 7 of RFC3168 already provides the means to address this concern, as it reads:

> First, ECN-Capable routers will only mark packets (as opposed to dropping them) when the packet marking rate is reasonably low. During periods where the average queue size exceeds an upper threshold, and therefore the potential packet marking rate would

> be high, our recommendation is that routers drop packets rather then set the CE codepoint in packet headers.
>
> Safe deployment of ECN requires that network devices drop excessive traffic, even when marked as originating from an ECN-capable transport. This is a necessary safety precaution because:..

Alternative behaviour. If we were to allow setting the ECT codepoint in the SYN packets, we need to define how it would behave.

One challenge is to support legacy ECN responders that do not support ECT marked SYNs but do support ECN.

One possible behaviour could be something along these lines. The SYN packet will carry the ECT(1) bit set as well as the ECE and CWR bits set. This is needed to support legacy ECN responders that would ignore the ECT bit, but properly process the ECN support negotiation using the ECE and CWR flags. Routers can then set the CE bit in the SYN.

If the responder receives a SYN with ECT(1), ECE and CWR bits set, it replies with a SYN/ACK that includes ECT(1) bit set. Because the ECT(1) bit is set, (and the CWR bit is not set) the initiator can realize that the responder supports ECN and also ECT marked SYNs.

If the responder receives a SYN with ECT(1), ECE, CWR and CE bits set, it replies with a SYN/ACK that includes the ECT(1) and the ECE bits set. Because the ECT(1) bit is set (and the CWR bit is not set), the initiator can realize that the ECE bit means that the CE bit was set in the SYN and then can react accordingly. The reaction to the ECE bit is then to halve the initial CWND for the connection.

4. Pure ACKs.

RFC3168 exposes the following arguments for not allowing the ECT marking of pure ACKs. In section 5.2 it reads:

> To ensure the reliable delivery of the congestion indication of the CE codepoint, an ECT codepoint MUST NOT be set in a packet unless the loss of that packet in the network would be detected by the end nodes and interpreted as an indication of congestion.
>
> Transport protocols such as TCP do not necessarily detect all packet drops, such as the drop of a "pure" ACK packet; for example, TCP does not reduce the arrival rate of subsequent ACK packets in response to an earlier dropped ACK packet. Any proposal for extending ECN- Capability to such packets would have

> to address issues such as the case of an ACK packet that was marked with the CE codepoint but was later dropped in the network. We believe that this aspect is still the subject of research, so this document specifies that at this time, "pure" ACK packets MUST NOT indicate ECN-Capability.

Later on, in section 6.1.4 it reads:

> For the current generation of TCP congestion control algorithms, pure acknowledgement packets (e.g., packets that do not contain any accompanying data) MUST be sent with the not-ECT codepoint. Current TCP receivers have no mechanisms for reducing traffic on the ACK-path in response to congestion notification. Mechanisms for responding to congestion on the ACK-path are areas for current and future research. (One simple possibility would be for the sender to reduce its congestion window when it receives a pure ACK packet with the CE codepoint set). For current TCP implementations, a single dropped ACK generally has only a very small effect on the TCP's sending rate.

We next address each of the arguments presented above.

The first argument is about lack of reliability while conveying congestion notification information when carried in pure ACKs. This is the specific instance for the pure ACK messages of the reliability argument discussed in Section 2. In some cases, the loss of pure ACKs is not detected by the endpoints, loosing the congestion notification information indadvertedly if it was to be carried in those packets. As we argued before, the bar for deciding if a packet can be marked with the ECT codepoint i.e. if it is suitable for carrying congestion notification information is that the congestion signal communication should be as reliable as dropping the packet. After all, the alternative of setting the CE bit in the packet is dropping the packet. So, the question is whether carrying congestion information in a pure ACK conveys the congestion information as reliably as when the pure ACK is dropped and it is obvious that the answer to that question is clearly yes. If the pure ACK carrying the ECT and the CE bits set is later dropped by the network, it will be essentially falling back to the use of drop as congestion signal.

The second argument exhibited in RFC3168 is the lack of means in the sender of the pure ACKs to reduce the load that is creating the congestion. Again, marking the pure ACKs with the ECT codepoint and allowing them to carry congestion notification information would be no worse than not doing so from this perspective (and it would be much more detrimental form the overall performance perspective). The sender of the pure ACKs will receive the echo of the congestion notification and it may be able to reduce the CWND of the connection.

If it happens to be only sending pure ACKs and no data and it can react reducing the rate at which data is being sent, it would not be worse in terms of congestion than in the case that the pure ACK is dropped.

So, overall, we believe that in terms of conveying and reacting to congestion, allowing to set the ECT (and the CE) flags in the pure ACKs is not worse than not doing so (and dropping the pure ACK), but in terms of performance, not ECT marking the pure ACKs is certainly detrimental.

5. Retransmitted packets.

RFC3168 does not allow setting the ECT codepoint in retransmitted packets. The arguments presented in the specification for supporting this design choice are the following ones (the text is quite long, not sure if we should keep it all):

> This document specifies ECN-capable TCP implementations MUST NOT set either ECT codepoint (ECT(0) or ECT(1)) in the IP header for retransmitted data packets, and that the TCP data receiver SHOULD ignore the ECN field on arriving data packets that are outside of the receiver's current window. This is for greater security against denial-of-service attacks, as well as for robustness of the ECN congestion indication with packets that are dropped later in the network.
>
> First, we note that if the TCP sender were to set an ECT codepoint on a retransmitted packet, then if an unnecessarily-retransmitted packet was later dropped in the network, the end nodes would never receive the indication of congestion from the router setting the CE codepoint. Thus, setting an ECT codepoint on retransmitted data packets is not consistent with the robust delivery of the congestion indication even for packets that are later dropped in the network.
>
> In addition, an attacker capable of spoofing the IP source address of the TCP sender could send data packets with arbitrary sequence numbers, with the CE codepoint set in the IP header. On receiving this spoofed data packet, the TCP data receiver would determine that the data does not lie in the current receive window, and return a duplicate acknowledgement. We define an out-of-window packet at the TCP data receiver as a data packet that lies outside the receiver's current window. On receiving an out-of-window packet, the TCP data receiver has to decide whether or not to treat the CE codepoint in the packet header as a valid indication of congestion, and therefore whether to return ECN-Echo indications to the TCP data sender. If the TCP data receiver

ignored the CE codepoint in an out-of-window packet, then the TCP data sender would not receive this possibly- legitimate indication of congestion from the network, resulting in a violation of end-to-end congestion control. On the other hand, if the TCP data receiver honors the CE indication in the out-of-window packet, and reports the indication of congestion to the TCP data sender, then the malicious node that created the spoofed, out-of- window packet has successfully "attacked" the TCP connection by forcing the data sender to unnecessarily reduce (halve) its congestion window. To prevent such a denial-of-service attack, we specify that a legitimate TCP data sender MUST NOT set an ECT codepoint on retransmitted data packets, and that the TCP data receiver SHOULD ignore the CE codepoint on out-of-window packets.

One drawback of not setting ECT(0) or ECT(1) on retransmitted packets is that it denies ECN protection for retransmitted packets. However, for an ECN-capable TCP connection in a fully-ECN-capable environment with mild congestion, packets should rarely be dropped due to congestion in the first place, and so instances of retransmitted packets should rarely arise. If packets are being retransmitted, then there are already packet losses (from corruption or from congestion) that ECN has been unable to prevent.

We note that if the router sets the CE codepoint for an ECN-capable data packet within a TCP connection, then the TCP connection is guaranteed to receive that indication of congestion, or to receive some other indication of congestion within the same window of data, even if this packet is dropped or reordered in the network. We consider two cases, when the packet is later retransmitted, and when the packet is not later retransmitted.

In the first case, if the packet is either dropped or delayed, and at some point retransmitted by the data sender, then the retransmission is a result of a Fast Retransmit or a Retransmit Timeout for either that packet or for some prior packet in the same window of data. In this case, because the data sender already has retransmitted this packet, we know that the data sender has already responded to an indication of congestion for some packet within the same window of data as the original packet. Thus, even if the first transmission of the packet is dropped in the network, or is delayed, if it had the CE codepoint set, and is later ignored by the data receiver as an out- of-window packet, this is not a problem, because the sender has already responded to an indication of congestion for that window of data.

In the second case, if the packet is never retransmitted by the data sender, then this data packet is the only copy of this data

> received by the data receiver, and therefore arrives at the data receiver as an in-window packet, regardless of how much the packet might be delayed or reordered. In this case, if the CE codepoint is set on the packet within the network, this will be treated by the data receiver as a valid indication of congestion.

There are essentially three arguments for not ECT marking retransmitted packets, namely, reliability, DoS attacks and over-reaction to congestion. We address all of them next in order.

About reliability, as described in Section 2, we believe that the bar should be that the congestion signal should be delivered as reliably as if it was a packet drop. So, if a retransmitted packet is dropped and this goes by unnoticed by the receiver, then the congestion signal expressed as a drop would be lost. The same applies to the congestion signal resulting from marking with ECT and CE the very same retransmitted packet which later is dropped.

About the possibility of DoS attacks, the protection against the DoS attack does not result from not allowing retransmitted packets to be ECT marked. If an attacker decided to launch such an attack, it would craft the packet with the ECT codepoint set. Effectively, the protection against the described DoS attack comes from the requirement that the receiver should not ignore the CE codepoint in out-of-window packets. We proposed to allow ECT marking of retransmitted packets, in order reduces the chances of it being dropped, but keep the requirement to ignore the CE codepoint in out-of-window packets.

Finally, the third argument is about over-reacting to congestion. Basically, if the retransmitted packet is dropped, the sender will not react again to congestion (it has reacted already when it generated the retransmitted packet). If the retransmitted packet is CE tagged instead of dropped, then the congestion signal will arrive again to the sender who could potentially react again to congestion. However, this should not happen as RFC3168 imposes the condition that a sender must only react once per window to the congestion signal and this should not be an exception to this rule.

6. Window probe packets

RFC3168 presents only the reliability argument for preventing setting the ECT codepoint in Window Probe packets. Specifically, it states:

> When the TCP data receiver advertises a zero window, the TCP data sender sends window probes to determine if the receiver's window has increased. Window probe packets do not contain any user data except for the sequence number, which is a byte. If a window

probe packet is dropped in the network, this loss is not detected by the receiver. Therefore, the TCP data sender MUST NOT set either an ECT codepoint or the CWR bit on window probe packets.

However, because window probes use exact sequence numbers, they cannot be easily spoofed in denial-of-service attacks. Therefore, if a window probe arrives with the CE codepoint set, then the receiver SHOULD respond to the ECN indications.

The reliability argument has been addressed in Section 2. dropping the window probe message in the case the conditions for the Silly Window Syndrome are on, basically implies that the sender will be stalled until the new Window Probe message reaches the receiver, which agains results in a performance penalty.

On the bright side, receivers should respond to ECN messages in these packets, so changing the behaviour should be less painful than for other packet types.

7. Security considerations

TBD, not sure if there is any.

8. IANA Considerations

There are no IANA considerations in this memo.

9. Acknowledgments

TBD

10. Informative References

[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>.

[RFC5562] Kuzmanovic, A., Mondal, A., Floyd, S., and K. Ramakrishnan, "Adding Explicit Congestion Notification (ECN) Capability to TCP's SYN/ACK Packets", RFC 5562, DOI 10.17487/RFC5562, June 2009, <http://www.rfc-editor.org/info/rfc5562>.

[I-D.briscoe-tsvwg-aqm-tcpm-rmcat-l4s-problem]
Briscoe, B., Schepper, K., and M. Bagnulo, "Low Latency, Low Loss, Scalable Throughput (L4S) Internet Service: Problem Statement", draft-briscoe-tsvwg-aqm-tcpm-rmcat-l4s-problem-02 (work in progress), July 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-01 (work in progress), June 2016.

[judd-nsdi]
Judd, G., "Attaining the promise and avoiding the pitfalls of TCP in the Datacenter", NSDI 2015, 2015.

[ecn-pam] Brian, B., Mirja, M., Damiano, D., Iain, I., Gorry, G., and R. Richard, "Enabling Internet-Wide Deployment of Explicit Congestion Notification", PAM 2015, 2015.

Authors' Addresses

Marcelo Bagnulo
Universidad Carlos III de Madrid
Av. Universidad 30
Leganes, Madrid 28911
SPAIN

Phone: 34 91 6249500
Email: marcelo@it.uc3m.es
URI: http://www.it.uc3m.es

Bob Briscoe
Simula Research Lab

Email: ietf@bobbriscoe.net
URI: http://bobbriscoe.net/