More Accurate ECN Feedback in TCP

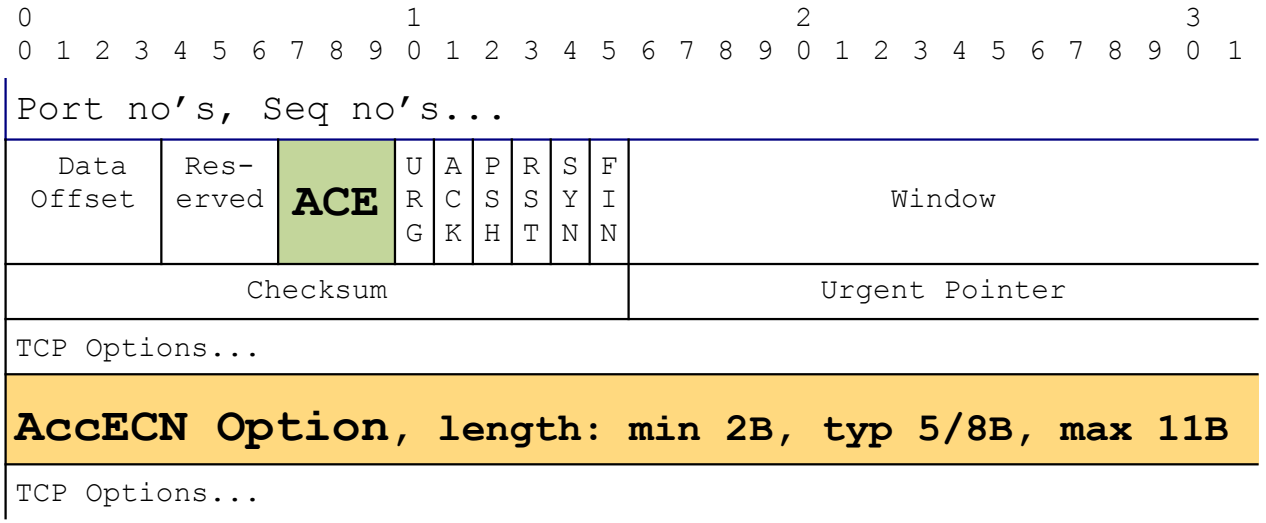
draft-ietf-tcpm-accurate-ecn-18

Bob Briscoe, Independent
Mirja Kühlewind, Ericsson
Richard Scheffenegger, NetApp

IETF-113 Mar 2022

Solution (recap)
Congestion extent, not just existence

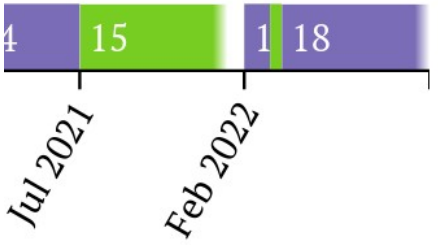
- AccECN: Change to TCP wire protocol
  - Repeated count of CE packets (ACE) - essential
  - and CE bytes (AccECN Option) – supplementary



- Key to congestion control for low queuing delay
  - 0.5 ms (vs. 5-15 ms) over public Internet

Recent draft history

draft-ietf-tcpm-accurate-ecn



- 12-Jul-21: -15
- 03-Feb-22: -16 [summary of diffs to list]
- 07-Mar-22: -17 [summary of diffs to list]
- 22-Mar-22: -18 [summary of diffs to list]
- Thank you for list discussion, mainly:
  - Ilpo on reconciling ACE field and AccECN TCP Option
  - Vidhi on response to mangling detection
  - Gorry on ACK filtering
  - Richard from experience implementing in FreeBSD
  - Bob noticing some unclear parts or errors
- Also off-list conversation with QUIC authors (Ian Swett & Jana) to align ACK frequency thinking

Implementation & Testing

- Linux implemention
  - recent changes to drafts still ToDo, otherwise...
  - will wait for IETF progress before submit to mainline
- FreeBSD implementation of draft-16 [RScheff]
  - added sending of AccECN TCP Options
  - omits handling arriving AccECN TCP Options
  - many cases to handle, but no unexpected problems
  - Open source
    - See https://reviews.freebsd.org/D21011 for AccECN
    - (Also https://reviews.freebsd.org/D23230 for ECN++)

Respond to ECN feedback even if not sending ECT packets?

- AccECN negotiated, but mangling detected
  - for the rest of the half-connection...

		Set ECT	Cong'n response?
a	Client (server) detects IP/ECN on SYN (SYN/ACK) mangled	N	Y
b	Data Sender detects continuous congest'n f/b	N	N
c	Data Sender detects ACE zeroed on 1st data pkt	N	N

  - Rationale (a):
    - mangling often at first hop – could be enabling ECT
    - then a subsequent bottleneck might be indicating genuine congestion
  - Rationale (b):
    - mangling most likely asserting CE – ignore and solely rely on loss
  - Rationale (c):
    - feedback from remote peer is suspect – ignore and solely rely on loss
    - even if combined with IP/ECN mangling, we don't believe RST would be useful
- Added as MUST (NOT)s in draft-17; draft-18 makes them advisory
  - 'cos best strategy might depend on deployment experience

Reviewed normative text throughout

- Reworded lower-case 'must', 'should', 'may' etc.
  - to 'needs to', 'ought to', etc.
  - so no-one can say maybe it was meant to be upper-case
- Upper-cased RECOMMENDED in two cases:
  - §1. RECOMMENDED to implement SACK & ECN++ with AccECN
  - §3.2.3 strongly RECOMMENDED to also test path traversal of the AccECN Option

Other changes

- See 5 spare slides for summaries of each diff in this IETF cycle (repeats of postings to list)
- The only technical changes:
  - Whether to respond to congestion if not sending ECT (see earlier slide)
  - §3.2.2.5.1 Increment-Triggered ACKs:
    'In either case, 'n' MUST be no greater than **7**.' (was 6)
  - §3.2.1 Made initialization of ECT(1) feedback counter **r.e1b the same as r.e0b**
    (different values was a hang-over from when we only had one type of TCP option)

Upcoming changes 1/2

- Interaction with ACK Filtering
  - Gorry not happy with updating RFC3449
    - ("TCP Performance Implications of Network Path Asymmetry")
  - Will try to change text to outline the problem and discuss possible way(s) forward, without recommending any changes and without updating RFC3449

Upcoming changes 2/2

- Switch round preferred partial implementation of AccECN Option?
  - Current:
    - Even if a developer does not implement sending of the AccECN Option, it is RECOMMENDED that they still implement logic to receive and understand any AccECN Options sent by remote peers.
  - Proposed:
    - Even if a developer does not implement logic to understand received AccECN Options, it is RECOMMENDED that they still implement logic to send AccECN Options to provide richer feedback to those remote peers that do understand it.
- Reasons:
  1) Originally believed that Data Receiver (which sends AccECN Options) would be the more complex side, but it's the simpler
  2) TCP Option needed more in upstream where ACK filtering is greatest
  3) Servers more likely to be Linux where TCP option already fully implemented; client OS's still to be implemented

Status & Next Steps

draft-ietf-tcpm-accurate-ecn-18

- Ready for WGLC, except
  - Check recent changes are OK
  - 2 upcoming changes (see prev 2 slides)
- draft-ietf-tcpm-generalized-ecn (EXP) dependent on AccECN

AccECN

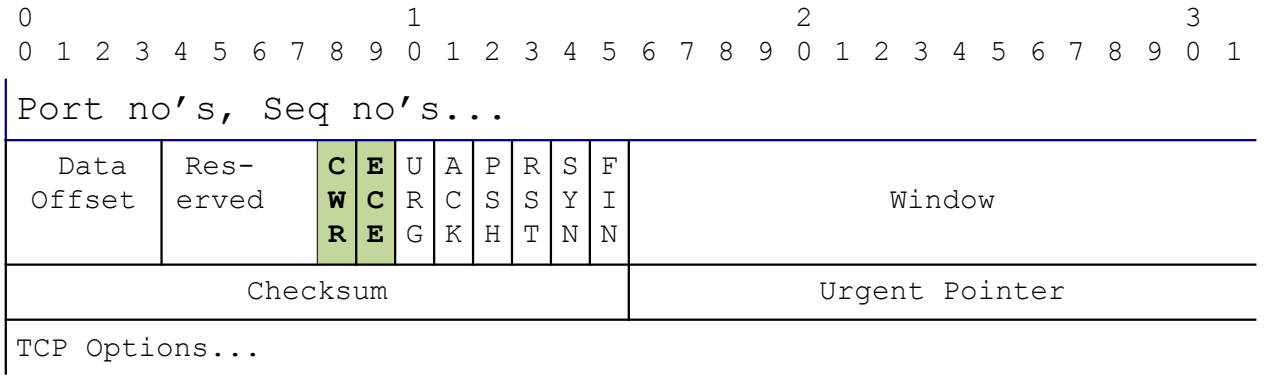
Q&A

spare slides

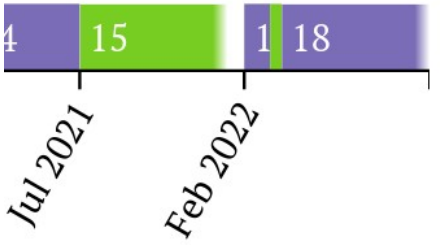
Problem (Recap)
Congestion Existence, not Extent

- Explicit Congestion Notification (ECN)
  - routers/switches mark more packets as load grows
  - RFC3168 added ECN to IP and TCP

IP-ECN	Codepoint	Meaning
00	not-ECT	No ECN
10	ECT(0)	ECN-Capable Transport
01	ECT(1)	
11	CE	Congestion Experienced



- Problem with RFC3168 ECN feedback:
  - only one TCP feedback per RTT
  - rcvr repeats ECE flag for reliability, until sender's CWR flag acks it
  - suited TCP at the time – one congestion response per RTT



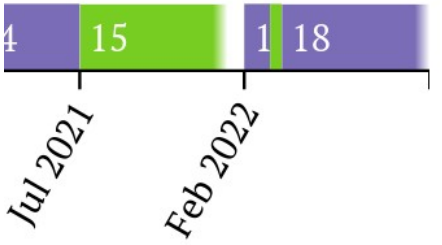
draft-16 03-Feb-22

draft-ietf-tcpm-accurate-ecn

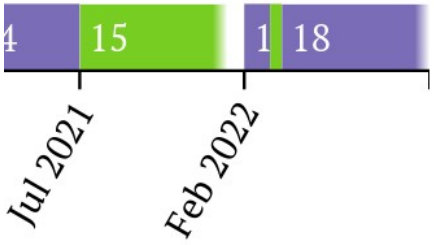
- Normative
  - §3.2.3.2.5 if using Option, MUST reconcile ACE with Option; previously only 'MUST consider...'.
    (Sets consistent baseline for future ACKs)
- Technical
  - A.2.1 Noted details not covered by pseudocode
- Editorial
  - §3.2.2.3 & §3.2.2.4 shifted section on test for mangling before test for zeroing ACE
    (To match likely execution order)

draft-17 07-Mar-22

draft-ietf-tcpm-accurate-ecn



- Normative:
  - §3.1.1 Put handshake requirements in execution order
  - §3.2 made "MUST increment byte counters" conditional on having implemented AccECN Option
  - §3.2.2.3 if continuous CE, added "**SHOULD NOT respond to CE feedback**" and "MUST remain in AccECN mode"
  - §3.2.2.5.1 Increment-Triggered ACKs:
    - "new data" means "newly delivered data"
    - 'In either case, 'n' MUST be no greater than **7**.' (was 6)
  - §3.2.3.3 SHOULD include an AccECN TCP Option if any byte counter has incremented (was 'if ACKs new data', but more clearly includes retransmissions)
  - §3.3.3 ACK filtering middleboxes 'SHOULD preserve the correct operation of AccECN feedback' (no longer suggests how)
- Technical & Editorial (next slide)



draft-17 07-Mar-22

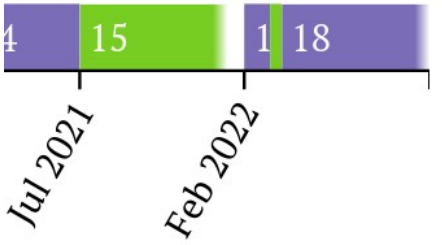
draft-ietf-tcpm-accurate-ecn

- Technical
  - §3.2.1 Made initialization of ECT(1) feedback counter **r.e1b the same as r.e0b** (different values was a hang-over from when we only had one type of TCP option)
- Editorial
  - Abstract: Called out the updates to other RFCs
  - Table of Contents: Increased depth to 4 (was 3)
  - Throughout:
    - Consistency with above changes
    - Added titles to all tables*
    - Other minor edits

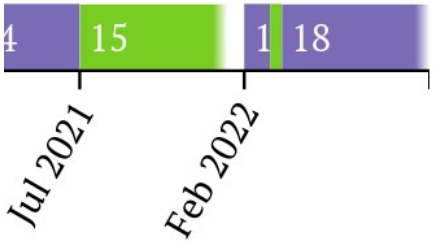
* xml2rfc now adds a 'Table xx' caption to the HTML rendering whether or not there is a caption title, and even if suppress-title=true is enabled

draft-18 22-Mar-22

draft-ietf-tcpm-accurate-ecn



- Normative
  - §2.5 Generic (Dumb) Reflector
    - Emphasized handshake reflection here is an example, not normative
  - §3.2.2.3 If either host detects IP/ECN mangling during handshake:
    - Advised not to send ECT packets but still respond to congestion f/b (was MUST NOT and MUST)
    - Added "MUST remain in AccECN mode"
  - If Data Sender detects continuous congestion f/b:
    - Advised not to send ECT packets and not to respond to congestion f/b (was SHOULD NOT and SHOULD NOT)
    - Added "MUST remain in AccECN mode"
  - §3.2.2.4 If Data Sender detects zeroing of ACE field after handshake
    - Advised **not to respond to congestion f/b**
  - Non-normative in all cases: 'cos depends on deployment experience



draft-18 22-Mar-22

draft-ietf-tcpm-accurate-ecn

- Normative (cont)
  - §1. RECOMMENDED to implement SACK & ECN++ with AccECN
  - §3.2.3 strongly RECOMMENDED to also test path traversal of the AccECN Option
  - (in both cases previously lower-case 'recommended')
- Editorial
  - Throughout: reworded lower case 'must', 'should', 'may', 'recommended', where it might be misinterpreted