ECN++: Adding ECN to TCP Control Packets

draft-ietf-tcpm-generalized-ecn-14

Bob Briscoe, Independent*

Marcelo Bagnulo, UC3M

TCPM WG, IETF-118, Nov 2023

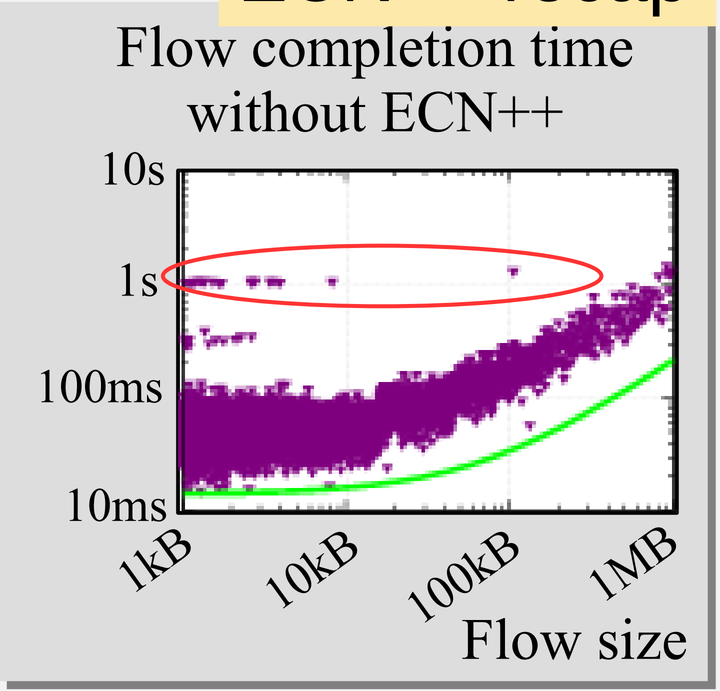
* Bob Briscoe's recent work on this document has been funded by Apple Inc.

ECN++ motivation

- Example: ECN-capable SYN
- Cuts flow completion time variance

- 1s timeouts: due to loss of TCP SYN or SYN/ACK
  - ECN++ protects TCP control packets from loss

ECN++ recap



<u>Experiment Details</u>
Each point represents FCT (SYN-FIN) of one ECN-Cubic flow over 7ms base RTT ADSL bottleneck @40Mb/s. With 2 long-running background flows. AQM: PIE in default config. Green line is ideal FCT if long-running flows were not present.

ECN++ sender (§3.2)

ECN++ recap

TCP packet type	RFC3168	ECN++ [draft-ietf-tcpm-generalized-ecn-14]		
		AccECN f/b negotiated	RFC3168 f/b negotiated	response to congestion experienced (CE)
SYN[1]	not-ECT	ECT[2]	not-ECT	[3]Reduce IW
SYN-ACK	not-ECT	ECT	ECT	Reduce IW
Pure ACK	not-ECT	ECT	not-ECT	[3] TBD for each CCA, .e.g. usual cwnd response and perhaps AckCC
Window probe	not-ECT	ECT	ECT	Usual cwnd response
FIN	not-ECT	ECT	ECT	None required, but could AckCC
RST	not-ECT	ECT	ECT	N/A
Re-XMT	not-ECT	ECT	ECT	Usual cwnd response
Data	ECT	ECT	ECT	Usual cwnd response

[1] For SYN, 'negotiated' means requested
[2] AccECN or equivalent safety, e.g. IW1 (client → server)
[3] Obviously only in AccECN case

Experiments can test any subset

ECN++ recap

ECN++ Forwarding & Receiving

Non-zero IP/ECN field on a TCP control packet or retransmission		
middlebox, eg. firewall	§3.1	RFC8311: "SHOULD NOT discard"
receiver (non-ECN++)	§3.3	SHOULD accept
receiver (ECN++)	§3.3	MUST accept

- §3.3 gives specifics for receiving each type of control packet, e.g.
  - SYN: if no logic to feed back CE, ignore and continue (ECN++ sender handles this safely)
  - Pure ACK: unless additional DupACK check on incoming pure ACKs, MUST NOT set ECT on outgoing pure ACKs (see later slide)
  - retransmission: if fails validity check, ignore CE
  - FIN: if fails validity check, ignore CE
  - RST: 'challenge ACK' [RFC5961] validity check recommended

ECN++ recap

Rationale (§4)

Rebuttals of main arguments in RFC3168

- Reliability argument
  - RFC3168: "MUST NOT set ECT on a packet if the loss of a CE mark [at a subsequent node] would be detected as an indication of congestion"
  - ECN++: "ECN is always more and never less reliable for delivery of congestion notification" (Do no extra harm)
- DoS Attacks
  - RFC3168: "ECN could be used to strengthen attacks, e.g. SYN flood"
  - ECN++:
    - **Sender**: Bad actors ignore prohibitions in RFCs, while good actors lose the benefits
    - **Network**: AQMs are already required to disable ECN when marking rate is high [RFC3168] [RFC7567]
    - **Receiver**: validity checks recommended [RFC5961]

Recent technical changes

draft-ietf-tcpm-generalized-ecn-12 → 14

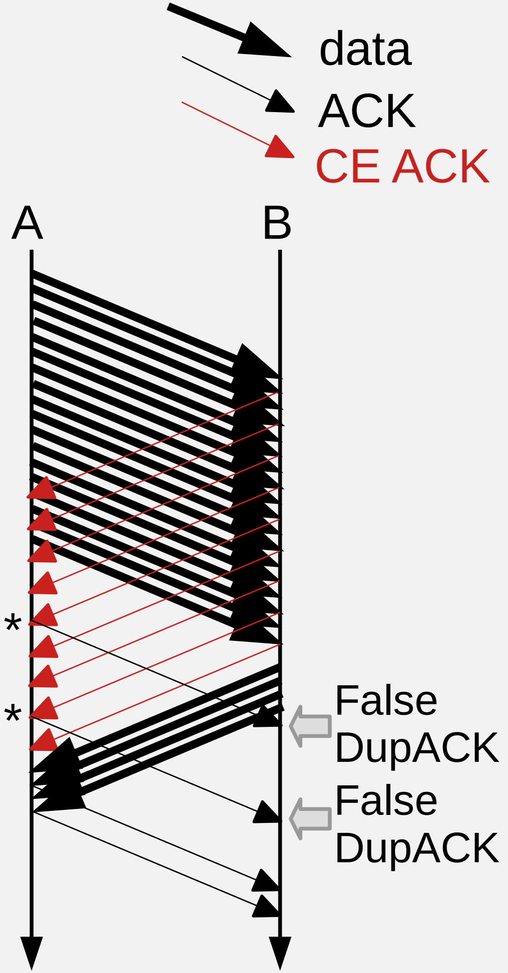
- Additional DupACK check
  - applicable to **all dup detection algos** (§3.3.3.1) [Markku - see next slide]
  - rewrote rationale (§4.4.4)
- Informative text about other transport protocols
  - updated summary of Not-ECT on SCTP control packets [draft-stewart-tsvwg-sctpecn] (§5.4)
- Security considerations:
  - easier fingerprinting of TCP stacks if each TCP implementation makes different control packets ECT (§6) [MScharf]

ACKs of ACKs

- Markku's concern
  - ACKs of ACKs can falsely appear to be DupACKs
  - could confuse algorithms that rely on DupACK detection (Limited Transmit, Fast Recovery, PRR, RACK-TLP etc) or other potential problems
- Solution adopted
  - **AccECN (stds track)** specifies ACK every 3 CE marked packets (*)

    could lead to ACKs of ACKs if sender sets ECT on pure ACKs, so:

    `"any spec that allows ECN-capable pure ACKs MUST require measures to distinguish ACKs of ACKs from DupACKs"`
  - **ECN++ (exp track)** gives 3 conditions for setting ECT on pure ACKs:
    - MUST have successfully negotiated SACK & AccECN
    - MUST apply check for dup incoming pure ACKs **in all dup detection algos**: if no SACK, despite SACK negotiated, not counted as dup
- Markku still concerned
  - absence of SACK might be due to 'A' supporting SACK but not DSACK
  - promised to explain impact on RACK-TLP and F-RTO by end of today



Recent editorial changes

draft-ietf-tcpm-generalized-ecn-12 → 14

- Updated numerous statements that said setting ECT is prohibited (by RFC3168) without mentioning that RFC8311 now allows it

- Fixed inconsistencies due to age of draft

Next Steps

- Ready for WGLC
  - now that AccECN has completed WGLC
- Please now review closely
  - esp. look for outdated text with fresh eyes