Accurate ECN Feedback for TCP (AccECN)

draft-ietf-tcpm-accurate-ecn

Bob Briscoe, Independent*

Mirja Kühlewind, Ericsson

Richard Scheffenegger, NetApp

TCPM WG, IETF-117, Jul 2023

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

TCP Congestion Feedback
Background & Problem

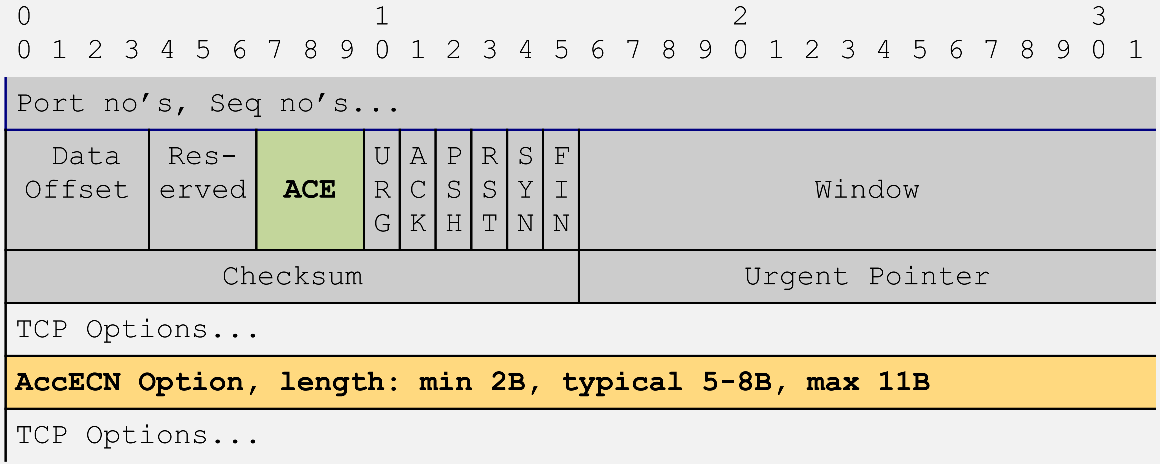
- DCTCP, L4S etc. repurpose standard ECN [RFC3168]
  - which "marks" more packets in the IP header (v4 & v6), the greater the queue.
- Sender keeps delay v low by adjusting rate in response to **extent** of marks
  - reported via end-to-end transport layer feedback

- Works fine with feedback in modern transports like QUIC or DCCP [RFC9000, RFC4340]
- ...but TCP was only designed to feed back existence not extent of congestion
  - sufficient when ECN was added back in 2001 [RFC3168]

Accurate ECN TCP feedback – recap

draft-ietf-tcpm-accurate-ecn

- AccECN reuses the 3 ECN flags in the main TCP header (ACE)
  - 3-bit counter to feed back number of IP-ECN marks
  - also to negotiate support by both TCP ends during the handshake



```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
```

Port no's, Seq no's...

Data Offset	Res-erved	ACE	URG	ACK	PSH	RST	SYN	FIN	Window

Checksum	Urgent Pointer

TCP Options...

AccECN Option, length: min 2B, typical 5-8B, max 11B

TCP Options...

- AccECN TCP Option optionally adds wider counters that rarely wrap

Recent Process History

- 24 Mar-23: 3-week WGLC #1 ended
- 30-Mar-23: draft-24 closed off WGLC issues
    - except waiting for Markku's response
- 24-May-23: Draft editor channelled 2 offlist issues to list
- 26-May-23: Markku's follow-up comments
    - mid-Jun-23: resolved or shifted text to ECN++ draft
- 30-Jun-23: Solutions to offlist issues all posted; no objections
- 10-Jul-23: Markku's repeat follow-up
    - response 22-Jul-23: see later slide
- 24-Jul-23: draft-25 (editorial Δs) and draft-26 (technical Δs) posted to close off WGLC#1
    - draft-25: editorial tidy up: consistency & moved 2 general req's from over-specific sections
    - draft-26: technical / normative issues resolved (?) – see following slides

Recent changes 1/6

draft-ietf-tcpm-accurate-ecn-25 to -26

- Markku Kojo's follow-on review [archive 26-May-23 1,2]
- Situation as of draft-24:
  - increment-triggered ACK rule:
    "**Receiver** MUST ACK 'n' CE marks" (including on ACKs)
    - even tho' ECN-capable ACKs are not stds track
    - for completeness and as mechanistic reflector
  - Only happens if **sender** makes pure ACKs ECN-capable
    - then conditional on SACK-negotiated, so that sender can distinguish non-SACK DupACKs
- Sender-side shifted out of AccECN spec to ECN++ (exp-track)
  - and generalized 'ECN++' to **any RFC for ECN-capable TCP control packets**
- Markku: Still potentially drags a stds-track receiver into experiments ack'ing ACKs
- Response: as a necessary generic stds-track receiver, not experimental
  - Analogy: experimental congestion controls use a generic stds-track TCP receiver

Recent changes 2/6

draft-ietf-tcpm-accurate-ecn-25 to -26

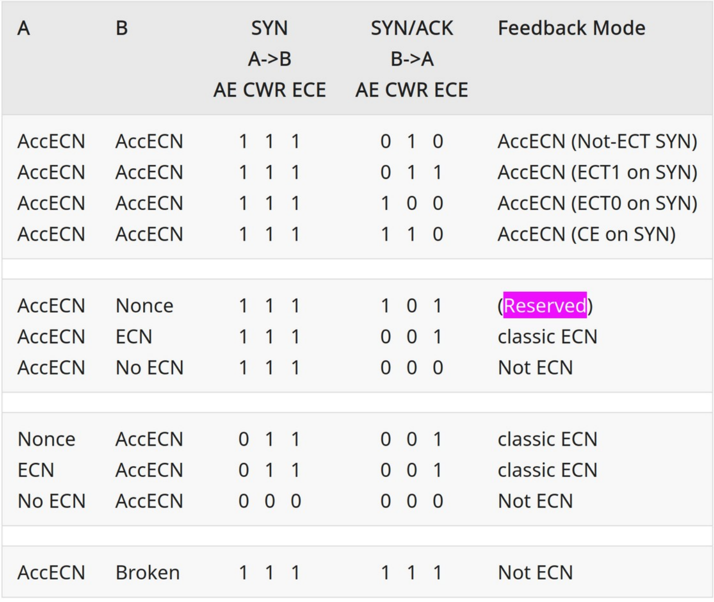
- Michael Tüxen as doc shepherd [archive 29-Jun-23]:
  - -24 Intro: AccECN RECOMMENDED to be implemented alongside SACK and ECN++
    - SACK is useful, but not essential
    - ECN++ benefits are not specific to AccECN
  - -25: AccECN RECOMMENDED to be implemented alongside SACK; TSopt can be useful
    - refs to SACK & TSopt made normative
    - moved ECN++ support text to §5.2 on compatibility with common options and current experiments

Recent changes 3/6

draft-ietf-tcpm-accurate-ecn-25 to -26

- Draft editor (offlist), with Wes Eddy's help
  - Generalized definition of 'acceptable packet' to work round IPR on RFC5961:
    ...passes tests in RFCs 9293 & 5961**, or other tests with equivalent protection**

- Draft editor [archive 19-Jun-23]
  - §3.1.3 Forward Compatibility: what AccECN does:
    - on receipt of any of the 5 reserved SYN combinations
    - but not on receipt of the reserved SYN/ACK combination
  - answer: treat as if negotiating AccECN
    - and consider that it feeds back that the IP/ECN field on the SYN was unchanged

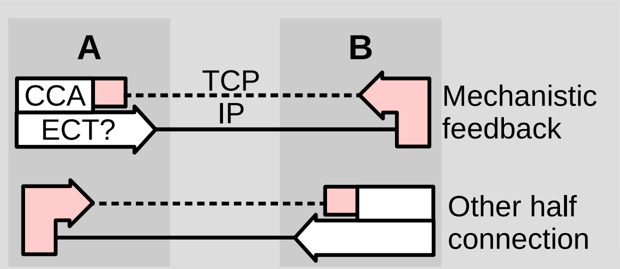


A	B	SYN	SYN/ACK	Feedback Mode
		A->B	B->A	
		AE CWR ECE	AE CWR ECE	
AccECN	AccECN	1 1 1	0 1 0	AccECN (Not-ECT SYN)
AccECN	AccECN	1 1 1	0 1 1	AccECN (ECT1 on SYN)
AccECN	AccECN	1 1 1	1 0 0	AccECN (ECT0 on SYN)
AccECN	AccECN	1 1 1	1 1 0	AccECN (CE on SYN)
AccECN	Nonce	1 1 1	1 0 1	(Reserved)
AccECN	ECN	1 1 1	0 0 1	classic ECN
AccECN	No ECN	1 1 1	0 0 0	Not ECN
Nonce	AccECN	0 1 1	0 0 1	classic ECN
ECN	AccECN	0 1 1	0 0 1	classic ECN
No ECN	AccECN	0 0 0	0 0 0	Not ECN
AccECN	Broken	1 1 1	1 1 1	Not ECN

Recent changes 4/6

draft-ietf-tcpm-accurate-ecn-25 to -26

- Unclear text found during Apple implementation [archive 19-Jun-23]:
- Handshake interaction with Classic ECN is fine, but fall-back to non-ECN lacks details (just says "based on RFC3168", which lacks details too)
  - Moved a couple of rules buried in over-specific sections:
    - if receive re-xmt of SYN or SYN/ACK, feed back the most recent IP-ECN field
    - after a feedback mode is entered, MUST NOT switch
  - Added example sessions and following general rules (§3.1.5):
    - a Server in AccECN feedback mode
      SHOULD emit AccECN SYN/ACK in response to non-ECN SYN
    - a Server in Non-ECN feedback mode
      SHOULD emit Non-ECN SYN/ACK in response to AccECN SYN
    - a **Server** in AccECN feedback mode
      MUST NOT set ECT, if it has received **or sent** a non-ECN SYN or SYN/ACK



  - Any AccECN implementation
    - SHOULD ignore TCP/ECN flags on SYNs & SYN/ACKs arriving after it's synchronized
  - Resulting rules checked against numerous patterns of loss & delay during handshake

Recent changes 5/6

draft-ietf-tcpm-accurate-ecn-25 to -26

- Draft editor [archive 30-Jun-23]:
  - 2 sections conflict: MUST/SHOULD test for zeroing of ACE field
    - §3.2.2.1: MUST test ACK of SYN/ACK:
    - §3.2.2.4: SHOULD test initial SYN=0 packet in either direction
  - Solution:
    - Exclude handshake packets from scope of latter section
    - Then 'SHOULD' → 'MAY' in latter section
      (if no zeroing during handshake, unlikely to be zeroing afterwards)

Recent changes 6/6

draft-ietf-tcpm-accurate-ecn-25 to -26

- IANA: TCP Option kinds [7 Nov 22]:
  - IANA assigned WG chair request: 172 & 174 (0xAC & 0xAE)
  - Error #1: I (BobB) wrote 172 & **173** into draft-22
  - Error #2: including the notes for IANA to publish about changing from experimental values
  - IANA faithfully copied these notes back to their experimental assignments page
- IANA noticed Error #1 [21 Jul 23]
  - implementations all (fortunately) followed IANA response: 172 & **174**
  - draft-26 now fixed to match main IANA page: 172 & **174**
  - editor notified IANA of Error #2 & IANA page fixed

Implementations status

- Apple client OS's [Vidhi Goel]:
  - AccECN TCP beta released 6 Jun'23 (off by default)
  - Developer resources: developer.apple.com/videos/play/wwdc2023/10004/
- FreeBSD [Richard S]
  - 14.0 reviews.freebsd.org/D21011
  - Plan to add AccECN TCP Options after 14.0 release reviews.freebsd.org/D36303
- Linux [Ilpo Järvinen, Neal Cardwell, Chia-Yu Chang]
  - against v5.15: github.com/google/bbr/commits/l4s-testing-2023-02-23-v3
    github.com/L4STeam/linux/blob/testing/README.md
  - mainlining in progress
  - RaspOS [Rob McMahon]
- Tools:
  - wireshark, packetdrill, tcpdump (wip) [Michael Tüxen, Richard S, Neal C, Vidhi G]

Next Steps

- This completes WGLC#1
- WGLC#2 after this meeting
- Thank you