

More Accurate ECN Feedback in TCP

draft-ietf-tcpm-accurate-ecn-22



Bob Briscoe, Independent



Mirja Kühlewind, Ericsson



Richard Scheffenegger, NetApp

IETF-115 Nov 2022

Recent draft history

draft-ietf-tcpm-accurate-ecn

- 20 → 21: 9 Nov '22
 - no changes (except formatting betw. v2 to v3 RFCXML)
- 21 → 22: 9 Nov '22 [[summary of changes on list](#)]:
 - if multiple SYNs, server MUST feed back latest IP-ECN
 - discovered missing from spec during Hackathon testing
 - recorded early IANA registrations of TCP Option Kinds

AccECN Roadmap

- Recap of AccECN landscape prior to WGLC (next 8 slides):
 - goal & approach
 - relation to other activities
 - placement in the stack
 - aspects to be reviewed
 - implementation status

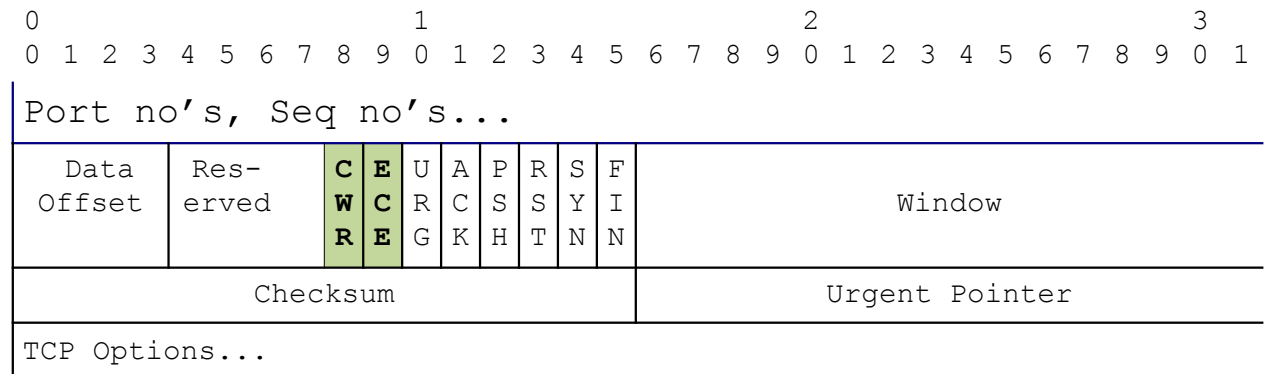
Goal

- Feed back extent of congestion not just existence
- To enable congestion control for very low queuing delay
 - 0.5 ms (vs. 5-15 ms) over public Internet

Problem (Recap)

Congestion Existence, not Extent

- 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



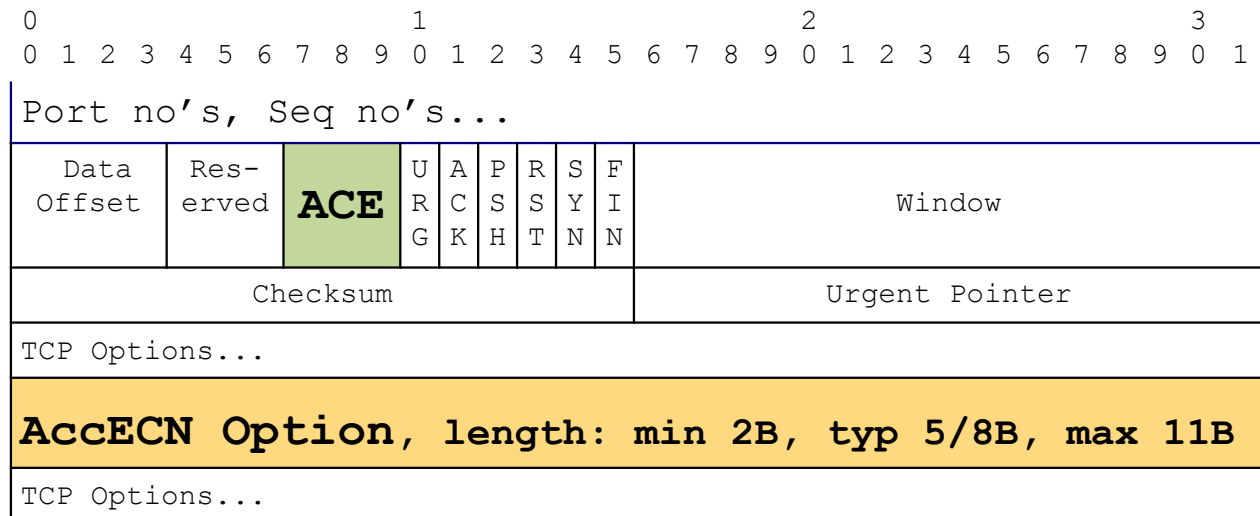
- Explicit Congestion Notification (ECN) recap
 - routers/switches mark more packets as load grows
 - RFC3168 added ECN to IP and TCP

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

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

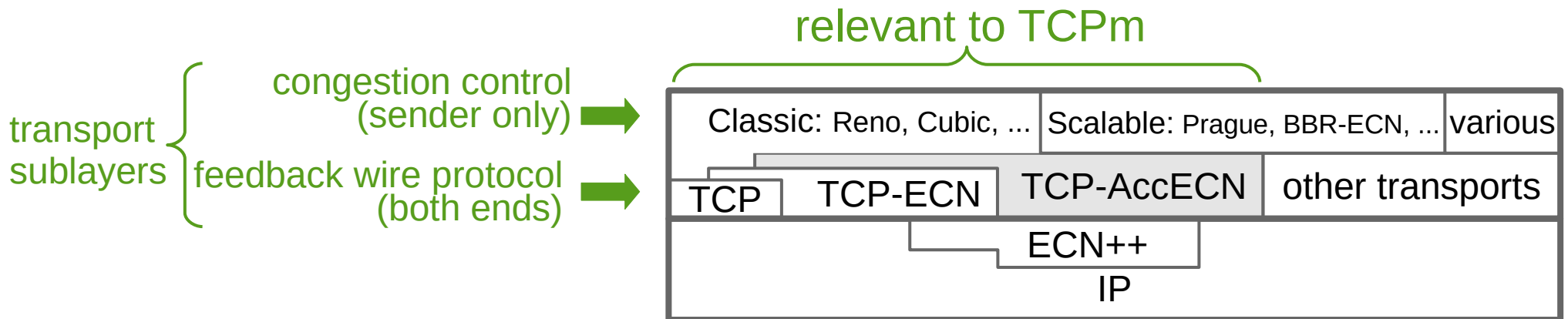


Relation to other activities

- DCTCP [RFC8257]
 - DCTCP's ECN feedback differs from RFC3168 and AccECN
 - but without negotiation (assumes DC-wide sys-admin)
 - Can use AccECN negotiation, and either AccECN or DCTCP-style ECN feedback
 - depending on initial value of the 3 TCP-ECN flags after 3WHS
 - already in Linux implementation of AccECN
- New Congestion Control Algorithms (CCAs)
 - AccECN steers clear of saying anything about congestion response
 - ECN feedback is wire protocol – architecturally 'below' a CCA
 - Can use AccECN ECN feedback for any CCA incl. Classic (Reno, CUBIC, BBR, ...)
- L4S experiment [draft-ietf-tsvwg-l4s-arch, ecn-l4s-id, etc.]
 - Low Latency, Low Loss and Scalable throughput
 - L4S CCAs can be used with TCP or other transport protocols (QUIC, etc)
 - L4S CCA with TCP requires AccECN to be negotiated
- ECN++ experiment [draft-ietf-tcpm-generalized-ecn]
 - Removes the rule against using ECN capability in IP-ECN of TCP control pkts & re-xmts
 - AccECN spec RECOMMENDS ECN++
 - Full capabilities of ECN++ only available with AccECN

Where AccECN Fits

- Can only enable AccECN if both TCP endpoints support it
 - falls back to RFC3168 TCP-ECN otherwise
 - no dependency on network changes
- Replaces & extends feedback part of TCP-ECN [RFC3168]
- Out of scope for AccECN:
 - Not what sender puts in the IP-ECN field
 - Not sender's congestion response to the feedback



Note: diagram shows what works over what;
not how an implementation would be structured

Aspects of AccECN to review

- Negotiation phase
 - Backward & forward compatibility
 - Mangling detection
 - Fall-back contingencies
- Resilience against ACK loss / coalescing
- Implications of TCP wire protocol changes
 - Implications of middleboxes / offload
 - Implications for middleboxes / offload
- Interaction with TCP variants
 - time-stamp, window scaling, SACK, TCP-AO, TFO, MPTCP, ...
- Security
 - flooding attacks, feedback integrity, downgrade attacks? ...

AccECN implementation status 1/2

- Linux (thx to Neal Cardwell)
 - Intended as reference implementation of the whole spec
 - Based off v5.15 kernel:
<https://github.com/google/bbr/commits/l4s-testing-2022-10-14-v1>
(merge into L4S repo imminent)
 - Also latest packetdrill tests:
<https://github.com/google/bbr/commits/l4s-packetdrill-2022-08-21-v1>
- Free BSD (thx to Richard Scheffenegger)
 - will be in FreeBSD 14 (without optional TCP option)
 - remaining parts in progress:
 - heuristic for long runs of missing ACKs
 - some details of the TCP option
 - passes all packet drill tests, except consistency betw. ACE & TCP Option

AccECN implementation status 2/2

- Apple platforms (MacOS, iOS, etc, thx to Vidhi Goel)
 - reflector side implemented – off by default
 - enable with `net.inet.tcp.accurate_ecn sysctl`
- Testing of all the above (Linux, FreeBSD, MacOS) in 2nd IETF L4S interop (co-located with this IETF)
- tcpdump patches for AccECN submitted (thx Richard Scheffenegger)
- Wireshark 4.0 decodes AccECN, incl. TCP option (thx Michael Tuexen)

Status & Next Steps

draft-ietf-tcpm-accurate-ecn-22

- WGLC
- draft-ietf-tcpm-generalized-ecn (EXP)
 - also ready for WGLC but dependent on AccECN

AccECN

Q&A