

ECN++: Adding ECN to TCP Control Packets

draft-ietf-tcpm-generalized-ecn-06



Bob Briscoe, Independent
Marcelo Bagnulo, UC3M



IETF 109 Nov 2020

- Refreshed 05 → 06
- Minor changes
- In holding pattern, waiting for AccECN

More Accurate ECN Feedback in TCP

draft-ietf-tcpm-accurate-ecn-13



Bob Briscoe, Independent

Mirja Kühlewind, Ericsson

Richard Scheffenegger, NetApp



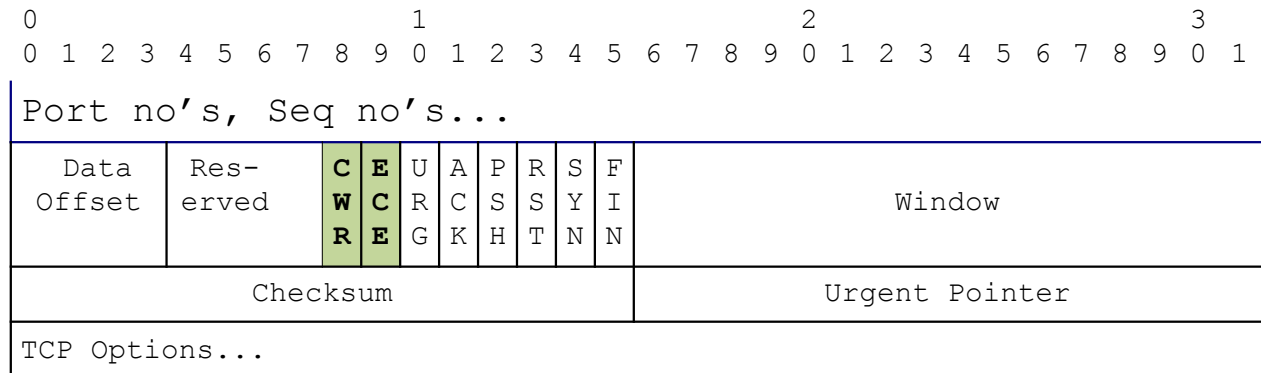
IETF 109 Nov 2020

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

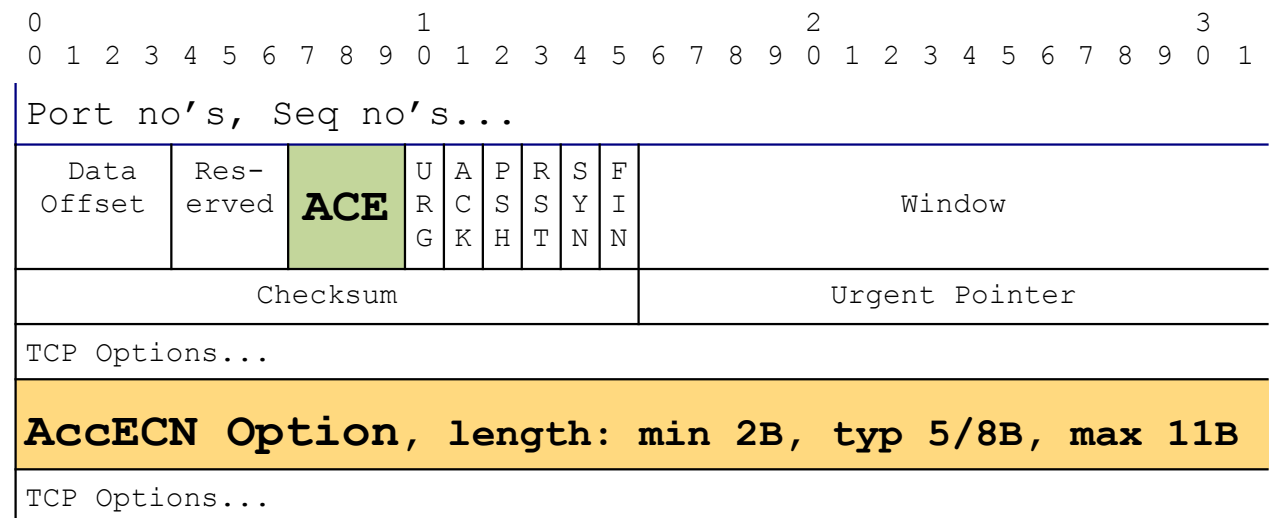


- Problem with RFC3168 ECN feedback:
 - only one TCP feedback per RTT
 - rcvr repeats **ECE** flag for reliability, until sender's **CWR** flag acks it
 - suited TCP at the time – one congestion response per RTT

Solution (recap)

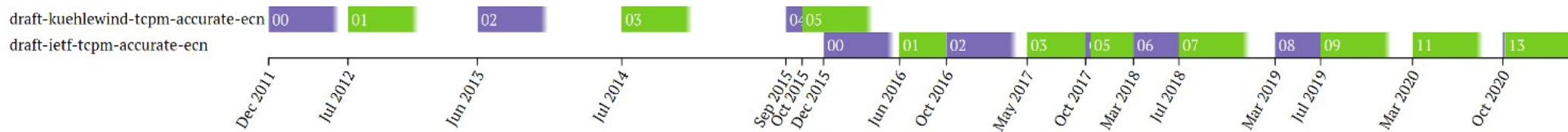
Congestion extent, not just existence

- 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
- Applicability: (see spare slide)

Activity since last update (Apr'20 interim)



- -11 to -12 (28 Oct '20):
 - Minor editorial fixes
- -12 to -13 (2 Nov '20)
 - Changed how to declare field order in AccECN TCP Option
 - SHOULD disable ECN if solid CE marking for a few rounds
 - Deeper and clearer recommendations for Proxies, offload engines and other middleboxes

Field Order of AccECN TCP Option

- How to distinguish 2 different field orders in the AccECN Option
 - ExxB = Echo Byte counter xx, where xx = E0, E1, CE (each 3 B)

kind0	length	EE0B	[ECEB	[EE1B]]
kind1	length	EE1B	[ECEB	[EE0B]]

- At IETF-109, two alternatives:
 - 1) Two Option Kinds [MScharf]
 - 2) Add flags byte to option [Ilpo]
 - No other proposals forthcoming
- Concern: Absence of a flags byte limits extensibility
 - But can we afford to burn 1B of option space on most packets?
 - Already 'forward compatibility' to add flags byte if needed
 - If length unrecognized, implementations MUST use as many 3-octet fields as fit
- Only choose the flags byte alternative, if prospect of other uses
 - Only one proposal, but logic for Ilpo's proposed 2-bit Cnt field seemed circular
- Conclusion: Two Option Kinds selected and written up

Testing for IP-ECN Mangling (§3.2.2.4)

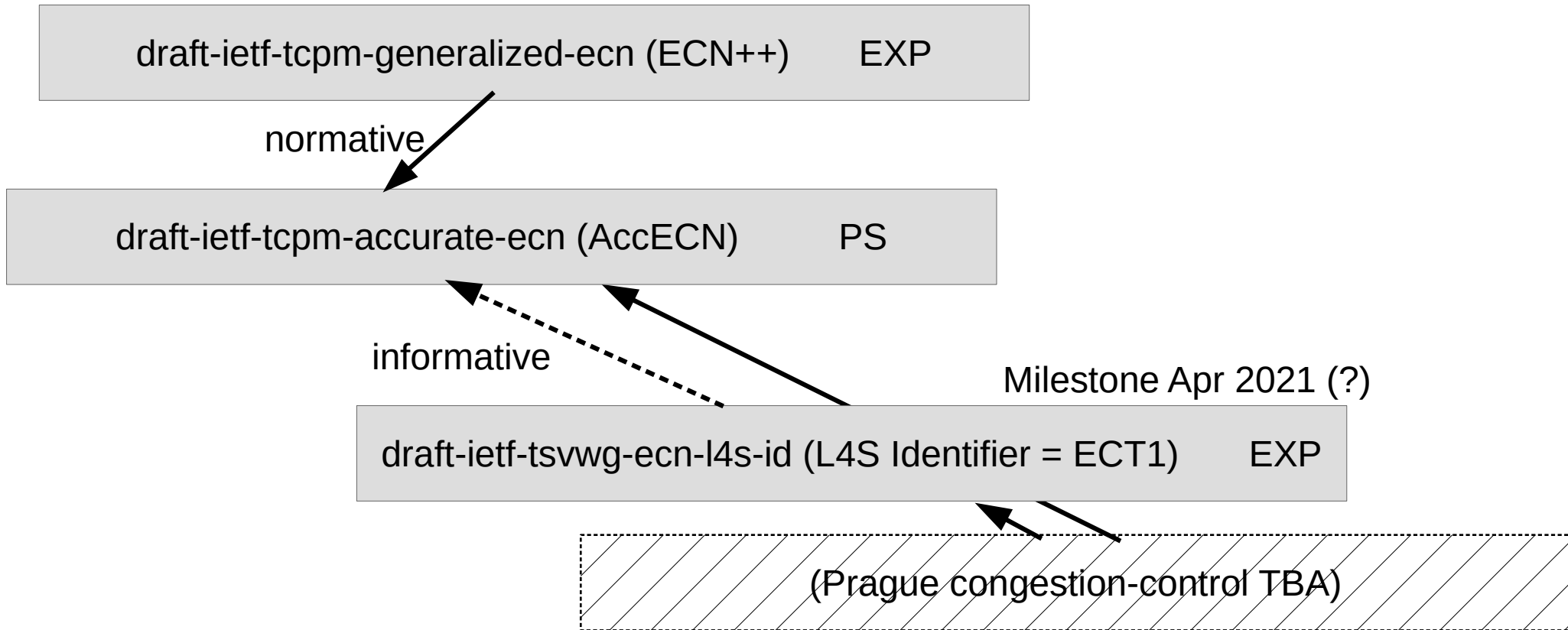
- Recommended additional test (paraphrased):
 - For first 3 or 4 rounds, AccECN Data Sender SHOULD check whether every packet it sent was CE-marked
 - If so, it SHOULD NOT send ECN-capable packets, but it MUST continue to feed back any ECN markings
- Already current practice for Classic ECN
 - in IoS, Linux, FreeBSD, at least

Recommendations for Middleboxes

Divided up existing section:

- Proxies (no change)
- Normalizers (no change)
- ACK Filtering
 - Made “SHOULD NOT coalesce” conditional on “If AccECN packet and middlebox can ECN mark”
 - Considerable list discussion
- Segmentation Offload
 - Described incremental deployment strategy
 - From today’s “Eject segment if ECN flags change at all”
 - To “Allow ACE field to change, but eject before wrap”

Draft dependencies



- ecn-l4s-id gives requirements for what L4S CC RFCs will have to say
- “Support for the accurate ECN feedback requirements [RFC7560] (such as that provided by AccECN [I-D.ietf-tcpm-accurate-ecn]) by both ends is a prerequisite for scalable congestion control in TCP.”

Status & Next Steps

draft-ietf-tcpm-accurate-ecn-13

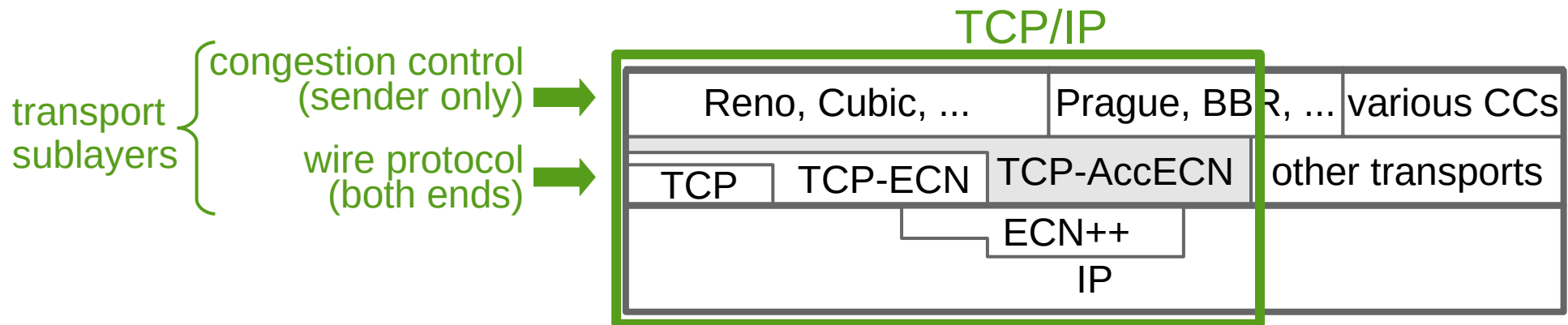
- Ready for WGLC
- draft-ietf-tcpm-generalized-ecn dependent on this
- April'20 tcpm interim:
 - WG resolved to wait a while for L4S, but go ahead soon if still waiting

AccECN

Q&A
spare slides

Where AccECN Fits

- Can only enable AccECN if both TCP endpoints support it ⁽¹⁾
 - but no dependency on network changes
- Extends the feedback part of TCP wire protocol
- Foundation for new sender-only changes (and for existing TCP), e.g.
 - congestion controls (TBA):
 - 'TCP Prague' for L4S ⁽²⁾
 - BBR+ECN
 - Full benefit of ECN-capable TCP control packets (ECN++) ⁽³⁾



(1) Backwards compatible handshake

- SYN: offer AccECN
- SYN-ACK can accept AccECN, ECN or non-ECN

(2) Low Latency Low Loss Scalable throughput [draft-ietf-tsvwg-l4s-arch]

(3) Without AccECN, benefit of ECN++ excluded from SYN [draft-ietf-tcpm-generalized-ecn]