More Accurate ECN Feedback in TCP

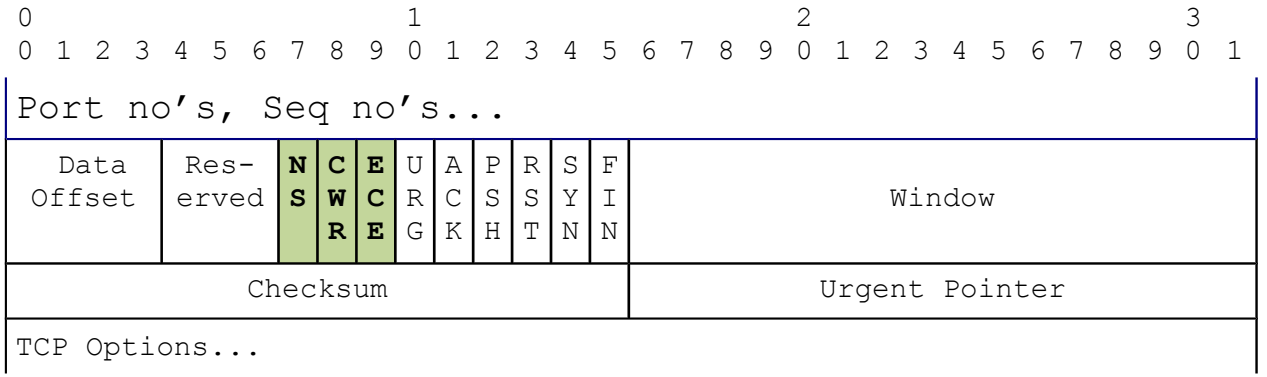
draft-ietf-tcpm-accurate-ecn-03

Bob Briscoe <ietf@bobbriscoe.net>
Mirja Kühlewind <mirja.kuehlewind@tik.ee.ethz.ch>
Richard Scheffenegger <rscheff@gmx.at>

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

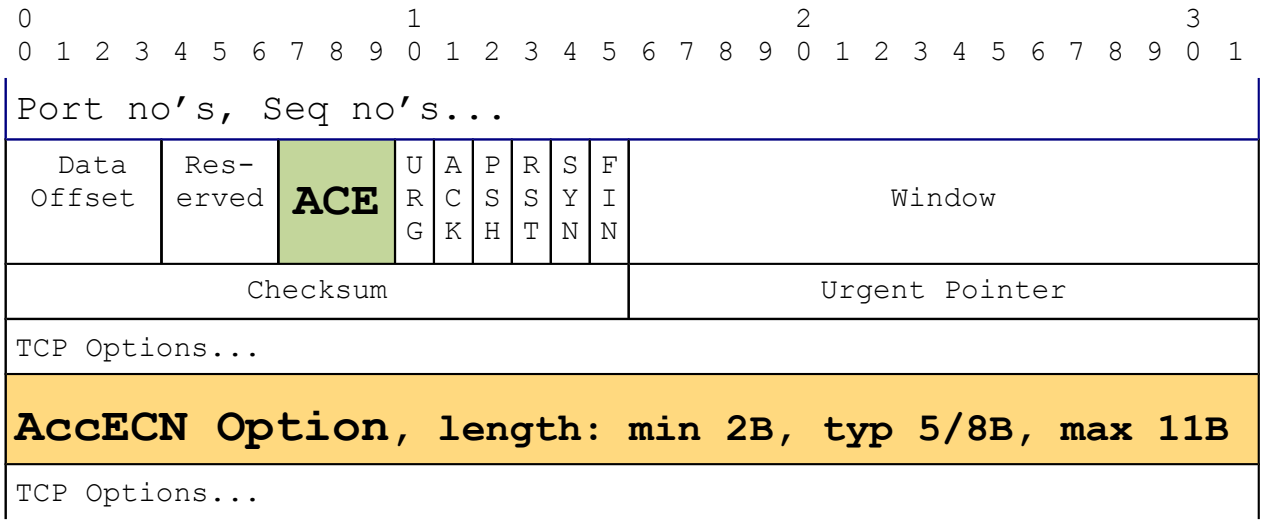
IP-ECN	Codepoint	Meaning
00	not-ECT	No ECN
10	ECT(0)	ECN-Capable Transport
01	ECT(1)	
11	CE	Congestion Experienced



- 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

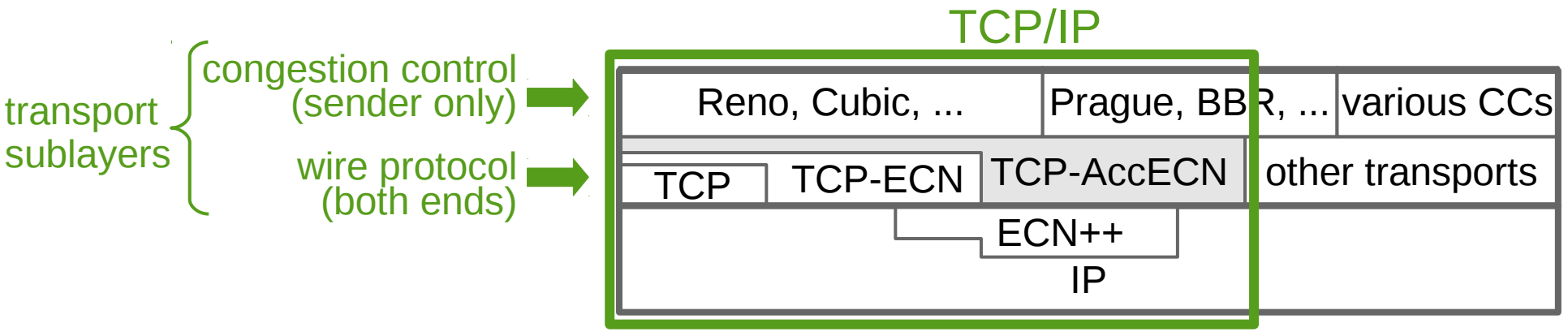
- AccECN: Change to TCP wire protocol
  - Repeated count of CE packets (ACE) - essential
  - and CE bytes (AccECN Option) – supplementary



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

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]

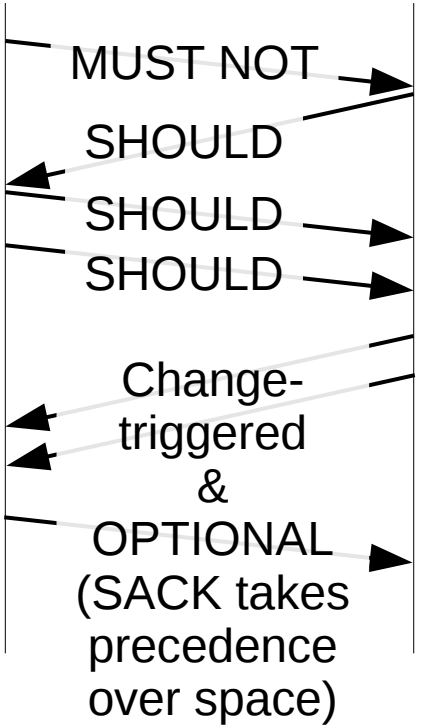
Recent Update – fall-back if bleached

A	B	SYN A->B			SYN/ACK B->A			Feedback Mode
		AE	CWR	ECE	AE	CWR	ECE	
AccECN	AccECN	1	1	1	0	1	0	AccECN
AccECN	AccECN	1	1	1	1	1	0	AccECN (CE on SYN)
AccECN	Nonce	1	1	1	1	0	1	classic ECN
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
AccECN	**AccECN+**	1	1	1	**0**	**1**	**1**	**AccECN (CU)**
AccECN	**AccECN+**	1	1	1	**1**	**0**	**0**	**AccECN (CU)**

- 2 unused handshake combinations (TCP ECN flags)
  - was: assume Non-ECN feedback
  - now: assume AccECN feedback
- Next rev: these are now needed to detect ECN bleaching
  - prevalent bug that wipes ECN – side effect of Diffserv bleaching
  - now that ECN++ is adopted (ECN on SYN)
    use these codepoints to feed back whether ECT(0/1) on SYN survived
- RFC3168 noted bleaching could happen, said it would be very bad, but silent on what to do about it (DISCUSS)

How Optional is the AccECN Option?

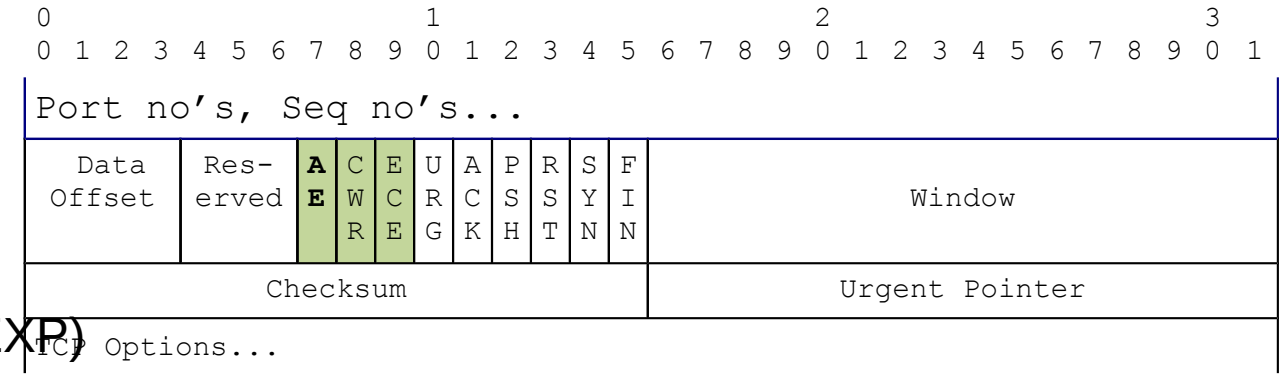
- AccECN Option:
  - has to be implemented
  - MUST NOT include on SYN (not needed)(1)
  - SHOULD(2) include on SYN-ACK, ACK and first client data segment
- Note: never a “MUST”
  - but have to try
  - nonetheless, no-one can prove you didn't



(1) AccECN negotiation in flags implies AccECN Option support

(2) not if cached as black-hole path

TCP NS flag → AE flag



- NS flag
  - currently assigned to ECN Nonce [RFC3540] (EXP)
- Registry policy for TCP flags is "Standards Action" meaning "a Standards Track RFC"
- AccECN is EXPerimental track
- Process to make RFC3540 historic is in progress [draft-ietf-ecn-experimentation] (PS) Submitted to IESG for Publication
- Two additional steps needed (agreed betw WG chairs in AD Office hours):

  1) IANA unassigns NS → reserved.
     write into IANA section of ecn-experimentation

  2) IANA assignment as AE:

     c) accurate-ecn assigns flag to itself, which needs the IESG to agree to this process exception

Status & Next Steps

- Implemented in Linux(1)
- Been waiting for:
  - NS flag to become available
  - ECN++ to be adopted (see item (A) below)

(1) https://github.com/mirjak/linux-accecn/

- Open Design Alternatives (see Appendix B)
  - A) Feed back all four ECN codepoints on the SYN/ACK (next rev)
  - B) Feed back all four ECN codepoints on the First ACK (DISCUSS)
- Open Issues (see Appendix C)
  1) Change-triggered ACKs: SHOULD or MUST? (DISCUSS)
  2) Is deliberate omission of AccECN Option a vulnerability?
  3) IANA Process
     - #2 can be left as part of the experiment

- Then ready for final reviews and WGLC

AccECN

Q&A

spare slides

Recent Updates

- Recent updates that impact implementation:
  - S.3.1.1: Forward compatibility with two unused combination of flags on the SYN/ACK (see earlier slide)
  - S.3.1.2: Minor changes to cache management for SYN timeout fallback
  - S.3.2.2: Tighter test for first segment in either direction, when checking initial value of ACE
  - S.3.2.5: Tighter AccECN Option traversal tests
  - 3.2.5.5. Consistency between AccECN Feedback Fields