

More Accurate ECN Feedback in TCP

draft-ietf-tcpm-accurate-ecn-11 & -10



Bob Briscoe, Independent

Mirja Kühlewind, Ericsson

Richard Scheffenegger, NetApp



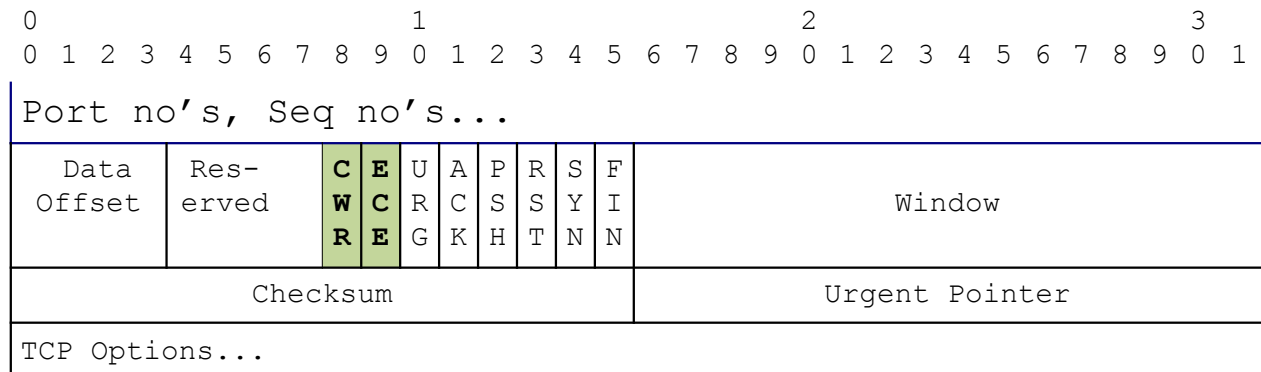
IETF interim-2020-tcpm-01 Apr 2020

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

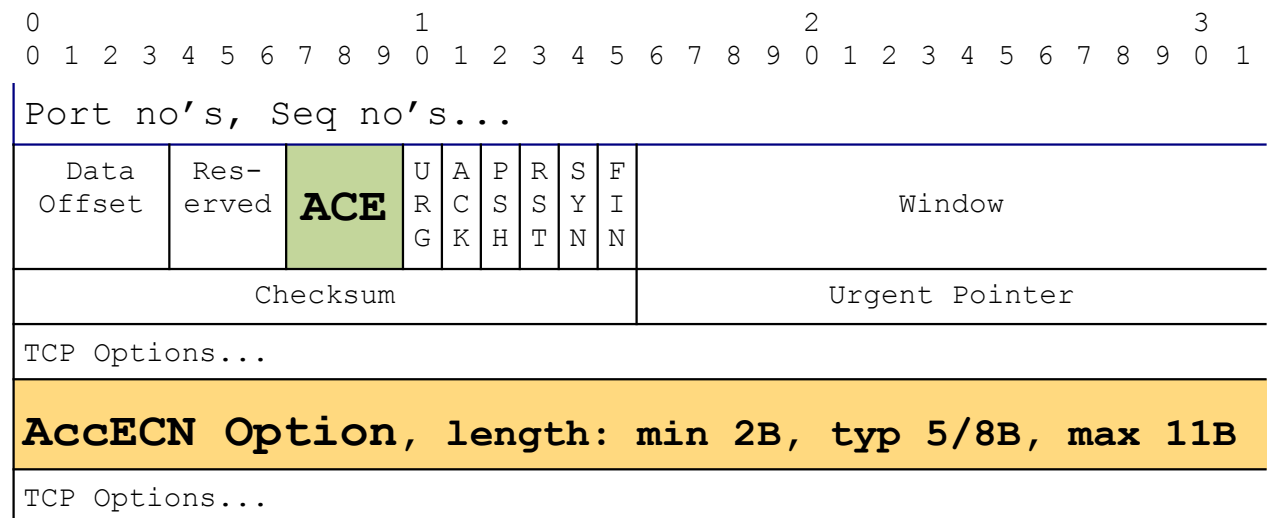


- Problem with RFC3168 ECN feedback:
 - only one TCP feedback per RTT
 - rcvr repeats **ECE** flag for reliability, until sender's **CWR** flag acks it
 - suited TCP at the time – one congestion response per RTT

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 status update (Nov'19)

- -09 to -10 numerous minor tech changes:
 - from list discussion since Nov-2019
 - niggles identified by Ilpo Jarvinen during Linux implementation for upstreaming
 - based on Olivier Tilman's, based on Mirja's
 - 6 main area covered on following slides:
 - Rights and obligations re. use of ECN
 - Backwards compatibility negotiation (tweaks)
 - Mangling Detection (tweaks)
 - Wrap of 3-bit ACE counter (tweak)
 - AccECN TCP Option (field order and usage)
 - Unusual Packet Arrivals
- -10 to -11 changes for exp → stds track

Changes 09 → 10 (Technical 1/6)

Rights & obligations re. use of ECN

- "Implications of AccECN Mode"
 - New section comparable to similar points in RFC3168

Data Sender in AccECN mode:

- can set ECT
- does not have to set ECT
- Congestion response
 - obliged to respond to CE f/b, as in RFC3168 as updated by RFC8311
 - MUST NOT set CWR on response

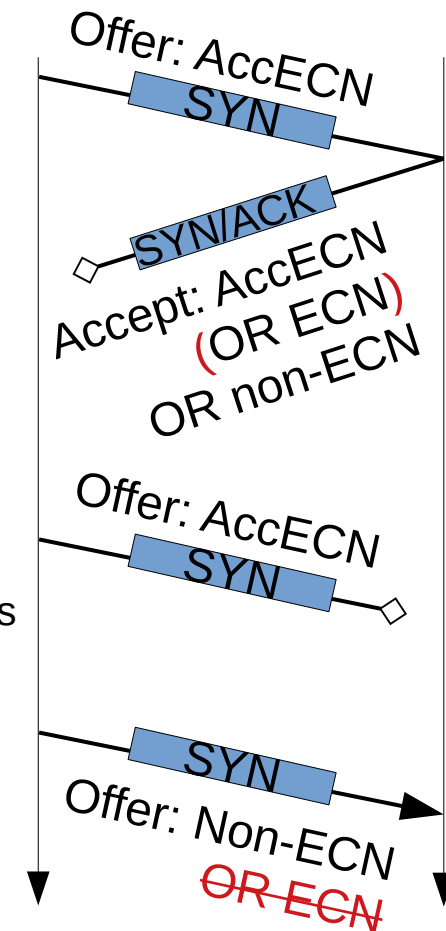
Data Rcvr in AccECN mode:

- MUST feed back IP-ECN as in §3.2
- if unwilling to send ECN feedback, should clear AE, CWR and ECE flags in SYNs and/or SYN/ACKs
- MUST NOT use reception of ECT in IP header as an implicit signal of ECN capability (could be due to mangling)

Changes 09 → 10 (Technical 2/6)

Backwards Compatibility Negotiation tweaks

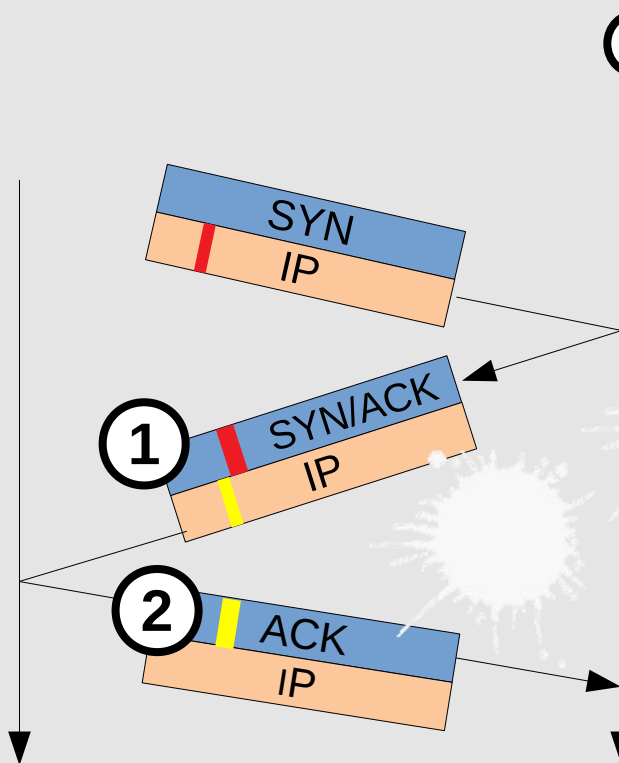
- AccECN server need not implement RFC3168 ECN (all clients still have to)
- Preclude mixed capability negotiation from either end
 - MUST NOT send SYNs or SYN/ACKs for both AccECN and RFC3168 ECN
 - If receive both, send RST
 - Reason: to prevent cases where each end's outcome after handshake could be inconsistent (in reordering corner-cases)
 - Implication: reduces freedom to choose SYN & SYN/ACK fall-back strategies
- Require retransmitted Fallback SYN to use same ISN
 - allows servers to detect ECN downgrade SYN attacks
- Reserved the codepoint combination used by the historic nonce case



A	B	SYN A->B			SYN/ACK B->A			Feedback Mode
		AE	CWR	ECE	AE	CWR	ECE	
AccECN	Nonce	1	1	1	1	0	1	(Reserved)

Mangling Detection Recap

Feedback of IP/ECN during 3WHS



①

A	B	SYN A->B			SYN/ACK B->A			Feedback Mode
AcceECN	AcceECN	AE	CWR	ECE	AE	CWR	ECE	
AcceECN	AcceECN	1	1	1	0	1	0	AcceECN (Not-ECT on SYN)
AcceECN	AcceECN	1	1	1	0	1	1	AcceECN (ECT1 on SYN)
AcceECN	AcceECN	1	1	1	1	0	0	AcceECN (ECT0 on SYN)
AcceECN	AcceECN	1	1	1	1	1	0	AcceECN (CE on SYN)

- Same coding on ACK

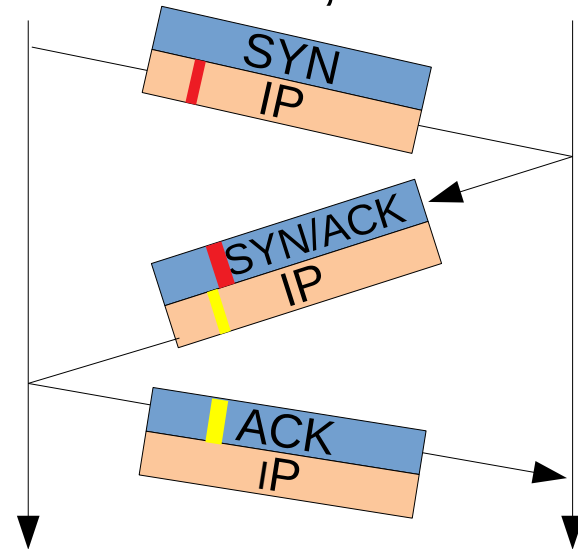
②

IP-ECN codepoint on SYN/ACK	ACE on pure ACK of SYN/ACK	r.cep of client in AcceECN mode
Not-ECT	0b 0 1 0	5
ECT(1)	0b 0 1 1	5
ECT(0)	0b 1 0 0	5
CE	0b 1 1 0	6

Changes 09 → 10 (Technical 3/6)

Mangling Detection Tweaks

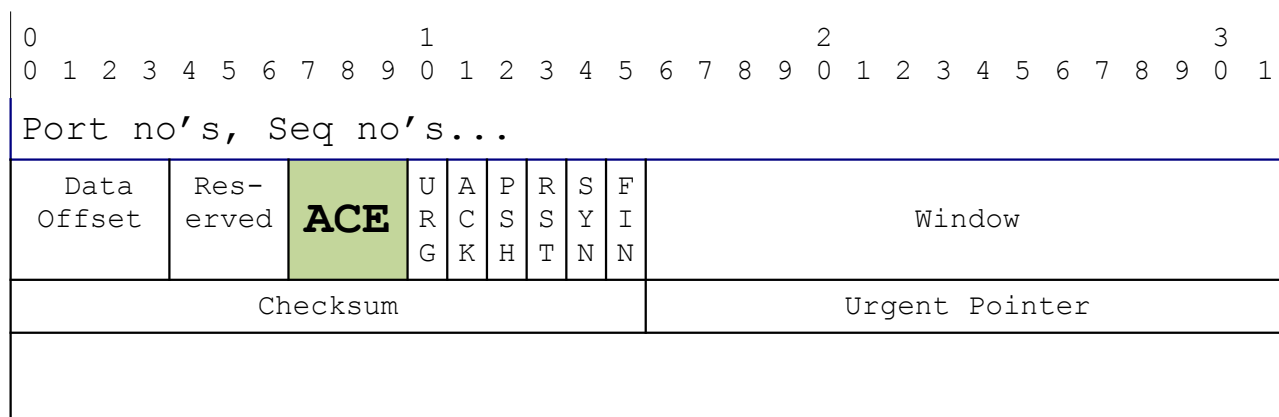
- Reflect IP-ECN field of SYN/ACK only on ACK of SYN/ACK, (not also on first data packet)
 - Reason: greatly simplifies implementation, esp with TFO.
 - repeating on first data packet was for reliable delivery, which is now achieved with ACE counter (see next bullet)
- Increment the ACE counter if CE on SYN/ACK but (still) not if CE on SYN
 - Reliable delivery of feedback of CE on SYN/ACK
 - Full mangling detection only unreliably delivered
 - Increment ACE no more than once (consistent with reflection on ACK)
- Redefine 'first packet' as first to arrive, not first in sequence in 2 cases:
 - Handshake reflection on the ACK of the SYN/ACK
 - In the test for zeroing of ACE
 - Reason: greatly simplifies implementation



Changes 09 → 10 (Technical 4/6)

Wrap of 3-bit ACE counter

- If ACE could have wrapped more than once, SHOULD assume “safest likely case”
 - not "conservatively assume" it did cycle
 - example algorithm in appendix
- Reason: avoid unnecessary hit on performance



Changes 09 → 10 (Technical 5/6)

AccECN TCP Option

- Allowed 2 different orders of the fields in the AccECN Option

kind	length	EE0B [init=1]	ECEB [init=0]	EE1B [init=0]
kind	length	EE1B [init=0x800001]	ECEB [init=0]	EE0B [init=0]

- Since consensus at IETF-107, Michael Scharf strongly disagrees; Alternatives:
 - 1) Two Option Kinds, or
 - 2) Add flags byte to option (see Ilpo's talk)
- More robustness (with flexibility) in rules including an AccECN Option
 - Change-triggered AccECN Option as SHOULD, not MUST
 - SHOULD follow change-triggered AccECN Option with another (removes ambiguity if ACK thinning or loss)
 - when same counter continues to increment, SHOULD consistently include it every n ACKs
 - Made rule about precedence of SACK conditional (max 2 SACK blocks)
 - MAY exclude counters that have not changed for the whole connection

Changes 09 → 10 (Technical 6/6)

Unusual Packet Arrivals

- Handled corner cases like In-window SYN during TIME-WAIT

Changes 09 → 10 (Editorial)

- Rationalized the structure and order of the sections
 - where the draft had evolved organically, some behaviours had been inserted in an irrelevant section, and others were repeated in two places
 - a number of the longer sections have been sub-sectioned to be clearer (and to be able to refer to specific aspects of the behaviour from other places)
- Added normative text for a number of the main behaviours (thx Ilpo)
 - obvious from the examples in the appendices, but not actually stated in the body.
- Acceptable Packets
 - Explicit about checking "acceptable packets"
 - before counting their ECN markings or before counting the ECN feedback they carry
- Caught text in one place that mentions a superseded behaviour in another
- Added reordering aspects to the summary of protocol properties
- Added to the justification for consuming header flags

Changes 10 → 11

- EXP track to STD track
 - Caught mentions of “experiment” throughout
 - Removed Experiment Goals section
 - New section “Updates to RFC 3168”

RFC3168	AccECN
§6.1.1 “TCP Initialization”	§3.1 “Negotiating to use AccECN”
§6.1.2. “The TCP Sender”	All stands except <ul style="list-style-type: none">• respond to counters not ECE• setting CWR no longer applies
§6.1.3. “The TCP Receiver”	§3.2 “AccECN Feedback”
§6.1.5. Acceptable re-xmt packet test	More stringent Acceptable Packet tests (for all packets)
§5.2, §6.1.1, §6.1.4, §6.1.5 and §6.1.6 prohibits use of ECT on ctrl pkts & rexmt	Requirements unchanged, but f/b defined, if such a pkt is not Not-ECT

Can reflection tests be removed?

- If mangling becomes a non-problem long-term
- Free up codepoints?
 - Would like to reduce from 4 to 2 reflection codes on SYN/ACK & on 3rd ACK
 - possible, but a drawn-out 2-stage process
 - burn another code:
Not CE = Not ECN || ECT0 || ECT1
then wait for use of the 3 old codes to subside
- Free up test processing?
 - Either end can just not check for a valid transition but they have to check for the CE transition anyway

Status & Next Steps

- Full implementation in Linux⁽¹⁾
 - patch (in 28 sequenced parts) submitted for upstreaming
 - on hold pending ECT(1) decision
- Implemented without TCP Option in FreeBSD⁽²⁾
- Ready for WGLC
as soon as tsvwg makes ECT(1) decision
- draft-ietf-tcpm-generalized-ecn in same holding stack

(1) <https://github.com/L4STeam/linux/tree/testing>

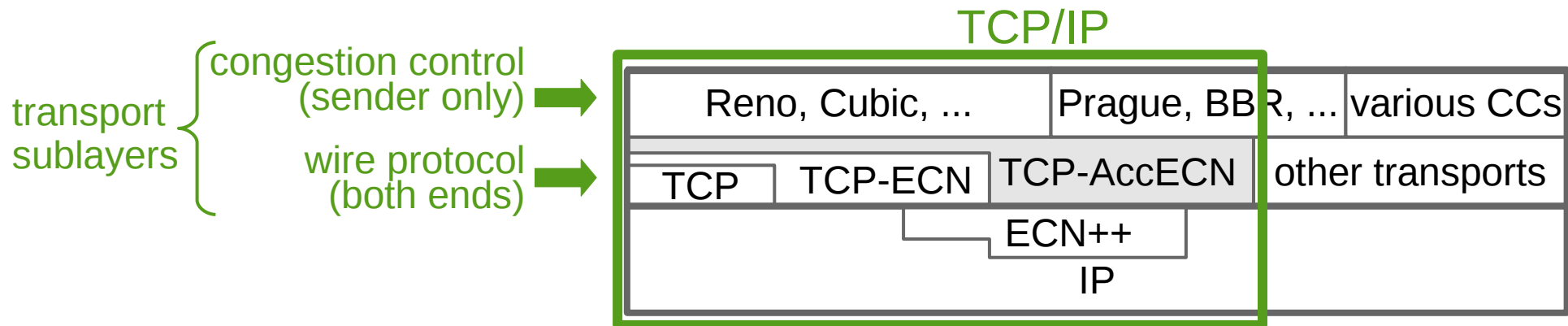
(2) <https://reviews.freebsd.org/D21011>

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]