More Accurate ECN Feedback in TCP

draft-ietf-tcpm-accurate-ecn-08

Bob Briscoe, CableLabs <ietf@bobbriscoe.net>
Mirja Kuhlewind, ERICSSON <ietf@kuehlewind.net>
Richard Scheffenegger, NetApp <rs.ietf@gmx.at>

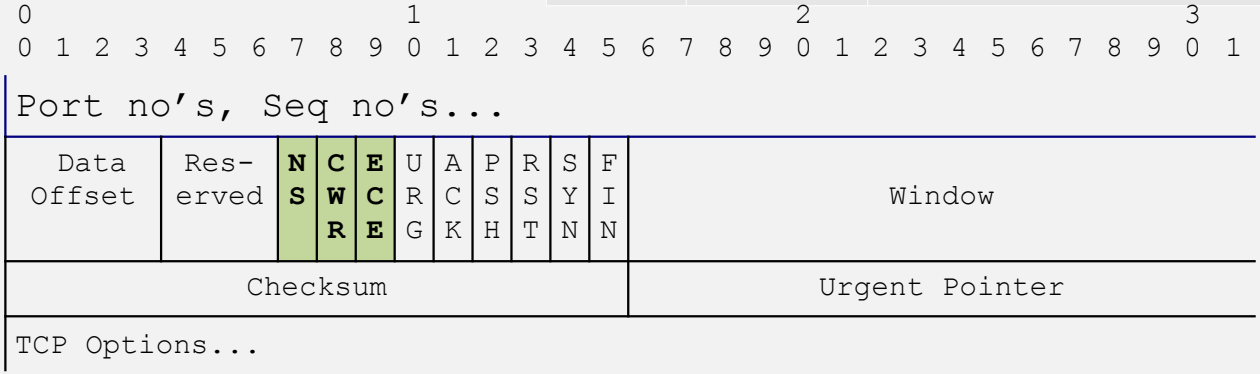


TCPM WG, IETF-104, Mar 2019

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

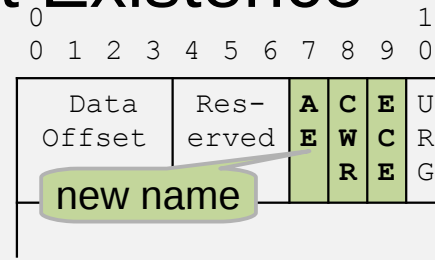
IP-ECN	Codepoint	Meaning
00	not-ECT	No ECN
10	ECT(0)	ECN-Capable Transport
01	ECT(1)	
11	CE	Congestion Experienced

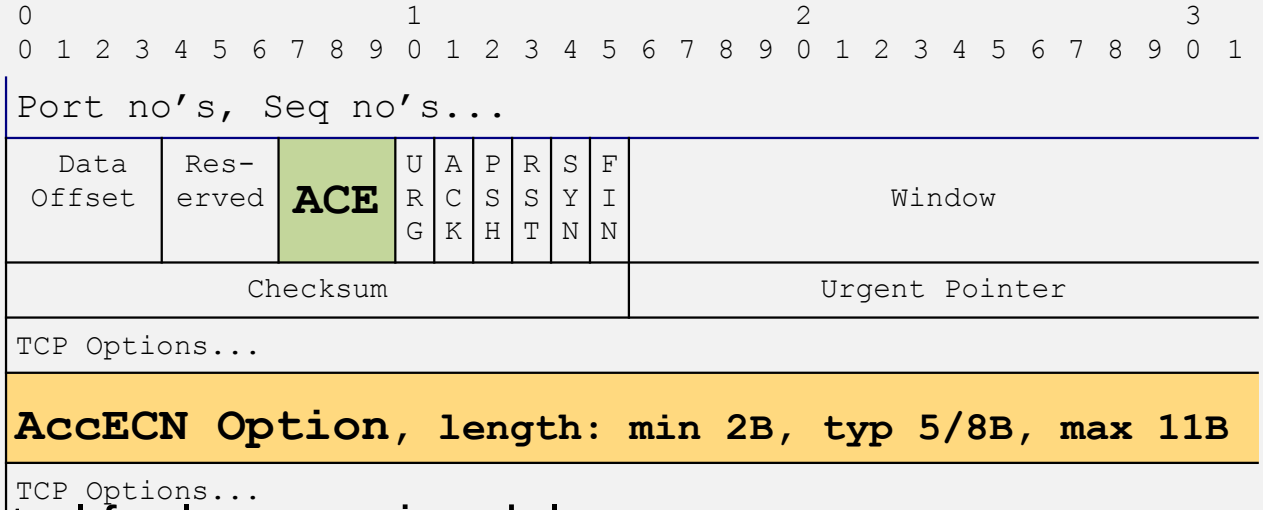


- Problem with RFC3168 ECN feedback:
  - only one TCP feedback per RTT
  - rcvr repeats ECE flag for reliability, until sender's CWR flag acks it
  - suited TCP at the time – one congestion response per RTT

Solution (recap): Congestion Extent, not just Existence

- AccECN: Change to TCP wire protocol
  - Repeated count of CE packets (ACE) - essential
  - and CE, ECT(0) and ECT(1) bytes (AccECN Option) – supplementary





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

Forward Compatibility

- Exhaustive check found more unused values
  - ECN TCP flags on SYN
- Solely about AccECN server behaviour
  - not a land-grab for AccECN
- Just an answer to the question
  - “if a future protocol uses any other combination on the SYN, which of the 3 possible server behaviours is likely to be most useful?”

AE	CWR	ECE	On SYN
0	0	0	Not ECN
0	1	1	RFC3168 ECN
1	1	1	AccECN
The other 5 combinations			AccECN server behaves as AccECN

- Makes behaviour from AccECN servers predictable for future protocols

Optional to implement the option

- protocol has to cope without it
- RECOMMENDED to implement snd & rcv
- If not snd, rcv handling RECOMMENDED

The AccECN Option has to be optional to implement, because both sender and receiver have to be able to cope without the option anyway - in cases where it does not traverse a network path. It is RECOMMENDED to implement both sending and receiving of the AccECN Option. If sending of the AccECN Option is implemented, the fall-backs described in this document will need to be implemented as well (unless solely for a controlled environment where path traversal is not considered a problem). Even if a developer does not implement sending of the AccECN Option, it is RECOMMENDED that they still implement logic to receive and understand any AccECN Options sent by remote peers.

Segmentation/coalescing offload

- Yuchung/Google wants to use DCTCP-style feedback
  - AccECN addresses problems with DCTCP f/b
  - but DCTCP-style better for current GRO
- Stepping back (see draft)...
  - ECN feedback and coalescing intrinsically conflict
  - DCTCP step marking induces runs of on or off
    - fortunately complementary to coalescing
  - Ramp marking being investigated to improve responsiveness
    - DCTCP stuck with step marking, without solution to the intrinsic conflict
- Solution
  - hardware can optimize around an (experimental) standard
  - so keep AccECN as is
  - patch software offload
    - Linux TSO/GRO coded at Hackathon
  - hardware will follow
    - review volunteered
  - provide interim local-use variant for DCTCP-style f/b
    - available within AccECN negotiation

AccECN Implementation

- Linux
  - ported to latest v5.1 kernel and submitted to mainline (Olivier Tilmans + Mirja Kühlewind)
  - fall-backs TBA
  - Hackathon:
    - testing / debugging
    - TSO/GRO added

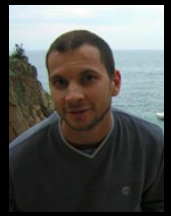
Status & Next Steps

- All the above is in draft-08
  - including resolution of Michael Scharf's issues
- Confirm GRO issue is resolved
- WGLC

- Some minor clarity edits from implementation experience
  - see mailing list – in authors' copy of draft-09

ECN++: Adding ECN to TCP Control Packets
draft-ietf-tcpm-generalized-ecn-03

Bob Briscoe, CableLabs® <ietf@bobbriscoe.net>
Marcelo Bagnulo, UC3M <marcelo@it.uc3m.es>



TCPM WG, IETF-104, Mar 2019

Bugfix: to prevent ECN++ disabling ECN on 84% of servers

- If a SYN requests ECN at the TCP layer and is already ECN-capable at the IP layer
  - Linux TCP listeners currently disable ECN for the connection
  - ECN++ client deployment hard to get started :(
- Recent tiny patch for back-porting to all Linux TCP listeners
  - identifies an ECN set-up SYN that's ECN-capable in IP by:

    `flag bits 4-9 == 0b000011`
  - not just

    `flag bits 8-9 == 0b11`
- This can distinguish an RFC3168 ECN setup SYN from something newer that allows ECT on a SYN, such as an AccECN setup SYN, which uses

  `flag bits 4-9 == 0b111`