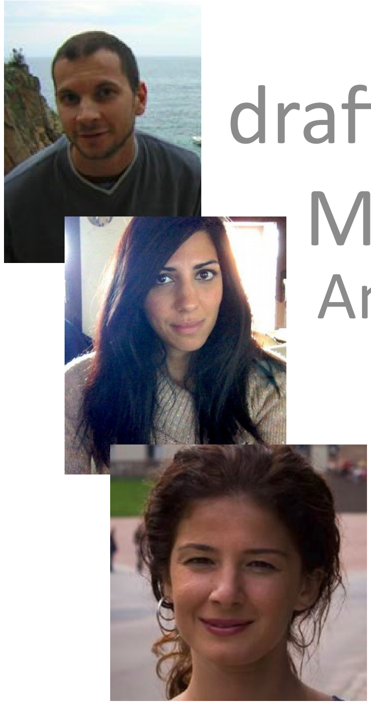
Adding Explicit Congestion Notification (ECN) to TCP control packets and TCP retransmissions

New name: ECN++

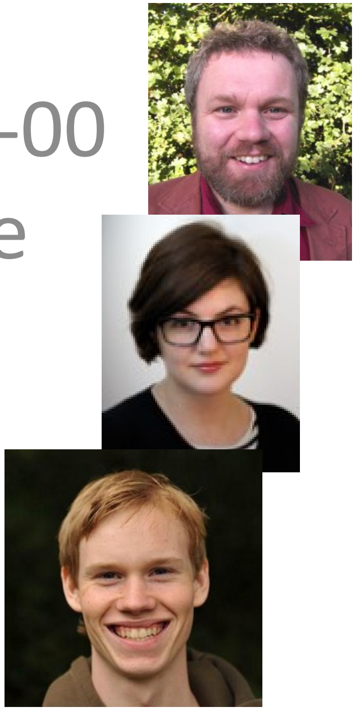


draft-ietf-tcpm-generalized-ecn-00

Marcelo Bagnulo, Bob Briscoe

Anna Maria Mandalari, Andra Lutu,
Özgü Alay and Henrik Steen

IETF-99, Jul 2017



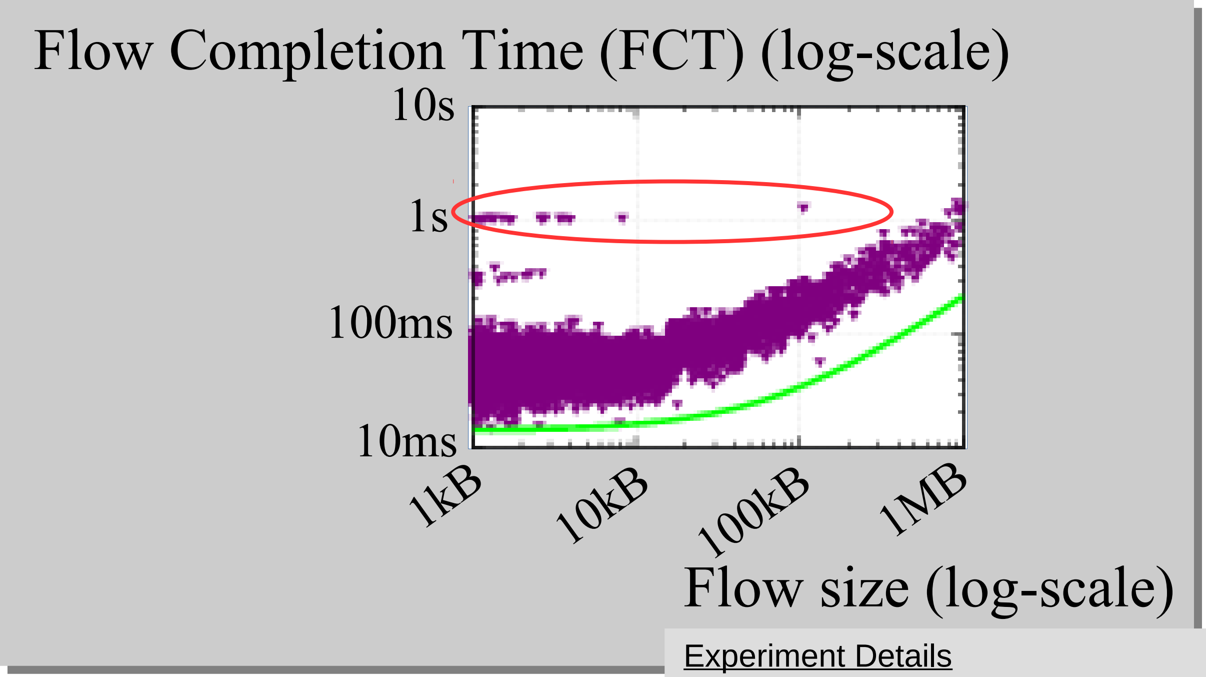
Recap – how we got here

- TCP/IP ECN spec [RFC 3168] argues against and prohibits using ECN on:
    - SYN
    - SYN/ACK
    - Pure ACK
    - Window Probe
    - Retransmission
- ECN+ and ECN+/TryOnce [RFC5562]: experiments to allow ECN on SYN/ACK

- ECN++ [draft-ietf-tcpm-generalized-ecn]
    - rebuts all the arguments
    - experiments with ECN on:
        - all the above, plus
        - FIN
        - RST
- ECN++ since Mar'17:
    - 5 fairly significant revisions
    - in response to 4 reviews
        - Mirja, DavidB, Padma, Gorry
    - recently adopted as TCPM work item

Main Benefits

1) Cut flow completion time variance (esp. short flows)
   - avoid initial retransmission timeout (default 1s)
   - otherwise SYNs & SYN/ACKs suffer background loss prob



Experiment Details

Each point represents FCT (SYN-FIN) of one ECN-Cubic flow over 7ms base RTT ADSL bottleneck @40Mb/s. With 2 long-running background flows. AQM: PIE in default config. Green line is ideal FCT if long-running flows were not present.

2) Re-xmt's etc. enjoy same ECN benefits as data

Where ECN++ fits

- Sender behaviour of each TCP half-connection
    - setting ECT, congestion response, fall-back
- TCP ECN feedback out of scope - compatible with both forms:
    - Standard [RFC3168]
    - More Accurate (AccECN) [draft-ietf-tcpm-accurate-ecn] (RECOMMENDED for primary benefit of ECT on SYN)

TCP packet type	AccECN f/b	RFC3168 f/b	Congestion response
SYN	ECT	not-ECT	Reduce IW
SYN-ACK	ECT	ECT	Reduce IW
Pure ACK	ECT	ECT	None or optionally AckCC [RFC5690]
Window Probe	ECT	ECT	Usual
FIN	ECT	ECT	None or optionally AckCC [RFC5690]
RST	ECT	ECT	N/A
Re-XMT	ECT	ECT	Usual

Optimistic ECT on SYN

Problem:

- Standard ECN [RFC3168] servers don't expect ECT SYN
  - so no space to feed back CE on SYN-ACK
  - AccECN SYN-ACK has space for CE feedback
- How does clients know which type of server?

Solution:

- Always request AccECN feedback (no cost)
- Always ECT on SYN (unless cache says no)
  - if SYN-ACK supports AccECN, also says CE or not
    - if CE, MUST reduce IW to 1 SMSS
  - else, no CE feedback logic at server
    - MUST conservatively reduce IW, SHOULD = 1 SMSS
    - OPTIONALLY cache

- Rationale for conservative IW reduction
  - client ⇨ server rarely uses IW > 1 (? DISCUSS)
  - even if it does, only one unnecessary IW reduction per non-AccECN server (then cached)
  - cache could also record non-AccECN proxy as an access network operator entry e.g. by checking a known AccECN test server

Response to CE on SYN-ACK

- Originally, draft just said “Follow RFC 5562”
  - but “ECN+/TryOnce” is over-literal interpretation of “ECN = drop”
  - attempts to mimic response to loss of initial packet
    (1s delay in case severely overloaded)
  - but CE implies network is delivering packets

- Approach: MUST reduce IW and SHOULD = 1 SMSS
  - same as original ECN+ scheme
  - authors of both (Aleks Kuzmanovic, Amit Mondal) agreed

Response to CE on a Pure ACK

- MUST reduce cwnd in response to any CE feedback
    - to regulate any data amongst the pure ACKs[1]
- MAY also implement AckCC [RFC5690]
    - to regulate pure ACK rate
- Is the receiver required to feed back CE on pure ACKs?
    - out of scope, see relevant feedback spec:
        - RFC 3168 is silent, so implementation-dependent
        - AccECN requires CE count to include CE on pure ACKs
- DISCUSS:
    - prohibit ECT on pure ACKs unless AccECN negotiated?
    - note:
      non-response to CE on pure ACK no worse than non-response to pure ACK loss

[1] RFC 3168 says
"Current TCP receivers have no mechanisms for reducing traffic on the ACK-path in response to congestion notification," which incorrectly assumes that one pure ACK implies that all the packets in the same flow are pure ACKs.

Response to CE on Re-XMTs

- Data receiver:
  - More stringent 'Challenge ACK' validity check [RFC5691] RECOMMENDED before feedback CE
  - mitigates blind out-of-window attack that fools the receiver into inducing a sender congestion response

Rationale for ECT on FINs, RSTs

- No congestion response possible
- Trade-off between
    - protecting FIN, RST from loss
    - strengthening FIN/RST attacks
- On balance, chosen to allow ECT
- ECT hardly strengthens flooding (see later)

Progress since Mar'17

Main technical deltas (see earlier slides)

- Clarified and tabulated where ECN++ fits [DavidB]
- Justified conservative response to CE on SYN if server not AccECN [Padma]
- SYN/ACK – no longer follows RFC5562 [consulted with co-authors]
- Pure ACKs: [Mirja & Padma]
  - Reduce cwnd & optionally AckCC [RFC5690]
  - Lengthy rationale for this approach added.
- Re-XMT, RST, FIN: RECOMMEND more stringent validity checks [RFC5691]

Main editorial deltas

- Motivation: current Internet as much as DC and L4S
- “Network SHOULD NOT treat ECT ctrl pkts differently” Moved to draft-ietf-tsvwg ecn-experimentation (PS)
- Structure
  - Specification: brief instructions (per packet type)
  - Rationale: arguments & rejected alternatives
  - clearly flagged “Measurement needed” paragraphs
  - clearly flagged fall-back sections (SYN, SYN-ACK)
  - General fall-back: new section [Padma & Gorry]
- TCP variants and derivatives: new section
  - IW10, TFO, L4S, SYN cookies,
  - ECN++ translates to SCTP, QUIC, etc. [Gorry]

Experiment I/2
ECN++ Flooding

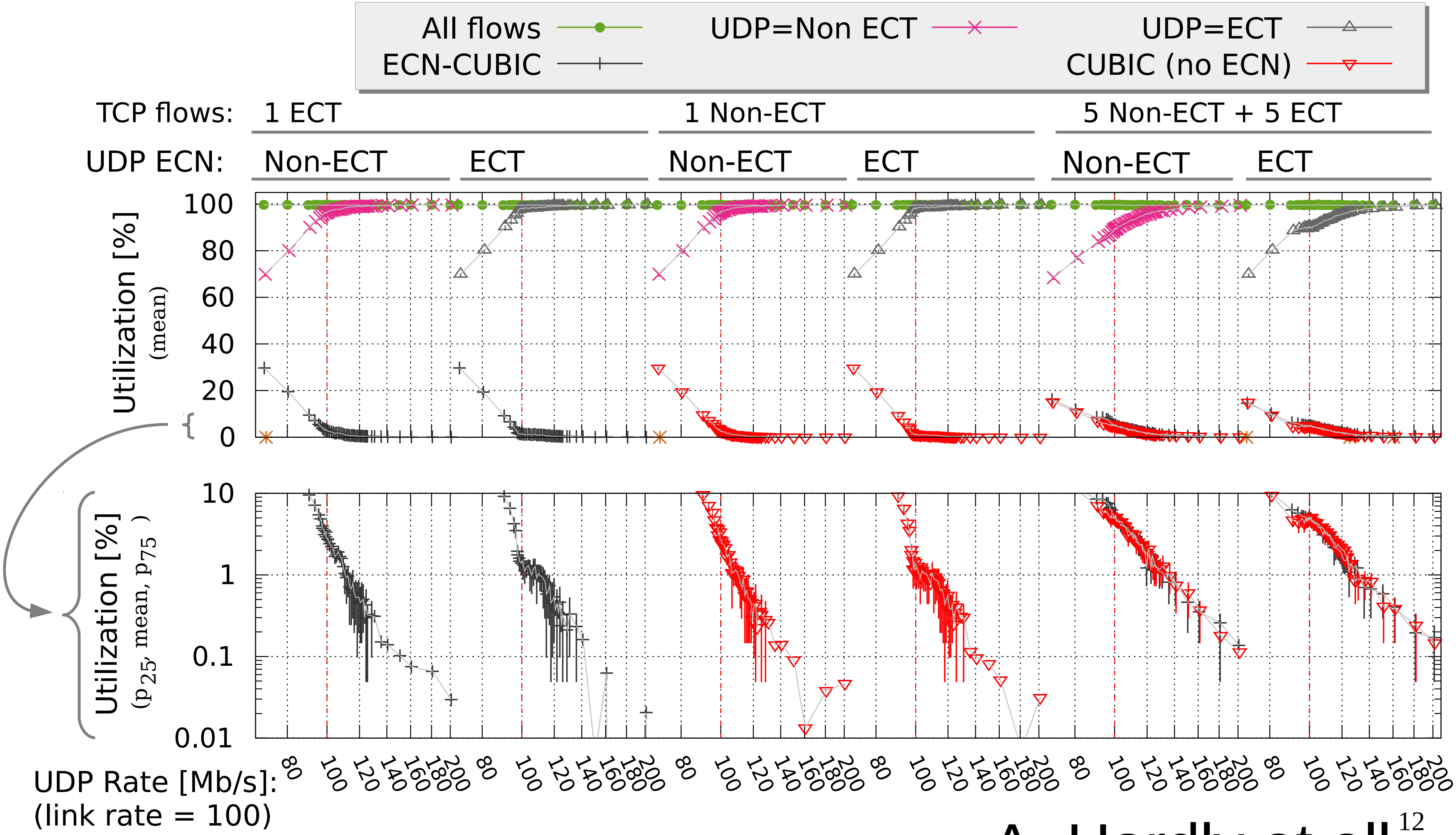
- “4.2.3. Argument 2: DoS Attacks” currently says

  “Further experiments are needed to test how much malicious hosts can use ECT to augment flooding attacks without triggering AQMs to turn off ECN support (flying "just under the radar"). If it is found that ECT can only slightly augment flooding attacks, the risk of such attacks will need to be weighed against the performance benefits of ECT SYNs.”

- We scanned the range of bit-rates to find whether and how much ECT attack packets can strengthen a flooding attack before the AQM turns off ECN

Q. Can ECN strengthen flooding?

- increase unresponsive attack from 70% to 200% of capacity. AQM: PIE (default config)
- spot the difference: with and without ECN



A. Hardly at all

Experiment 2/2
ECN++ Traversal

- Millions of measurements
  - from numerous TCP client vantage points on mobile and fixed networks
  - one-ended tests with Alexa 500k
  - two-ended tests with own servers
  - all allowed combinations of IP ECN field, for ECN and ECN++
  - all allowed combinations of TCP ECN flags, for ECN and AccECN
  - all types of TCP packet, in handshake and established connection
- Report under submission
- Main conclusion for ECN++
  - Wherever ECN got through, we found no problems for ECN++

Open issue

- fall-back on persistent loss of ECT control packets after connection establishment [Padma's & Gorry's review], e.g.
    - non-ECT SYN, then ECT on last ACK of 3WHS
    - route change to middlebox that black-holes ECT ctrl pkts
    - hard to detect
    - measurements so far (incl. mobile networks) show ECT ctrl pkts get through wherever ECT data gets through

Next steps

- ietf...-01 ready to post after this meeting
  - addresses Padma's and Gorry's review comments
  - other minor improvements
- ECN++ traversal experiments publication
- implementation pls
- more measurements pls
- review comments pls