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

draft-bagnulo-tcpm-generalized-ecn-00

M. Bagnulo & B. Briscoe

IETF97

# Proposed experiment

- Add ECN support to TCP control pkts and retx
  - SYNs, Wnd Probes, pure ACK, FINs, RSTs and retx
- Goals of the experiment:
  - Learn how the network and endpoints treat pkts marked with ECT(0), ECT(1) and CE
  - How much performance is improved by ECN support of these packets
  - Identify any issues with enabling ECN support of these packets, especially any security issues

# Specification: Network Behaviour
# Router or any middlebox

- if the router is not congested, the router SHOULD forward the packet.
  - SHOULD OR MUST?
    - SHOULD proposed to accomodate firewalls
- if the router is congested, then the router MAY set the CE codepoint in the packet instead of dropping the packet

# Endpoint behaviour: SYN

- Scope: only support ECN marking of SYNs for AccECN endpoints
  - The experiment does not support ECN marking of SYNs for RFC3168 endpoints, to preserve codepoints
- AccECN already defines how to feed back congestion notification when SYN is marked
- The proposed experiment is compatible and uses AccECN format.
- It completes the specification for its full support
  - Defines client behaviour. Server behaviour specified in AccECN

# Client endpoint behaviour: SYN

Sends SYN with ECT(0/1) and NS, ECE, CWR set

- Should also send a non-ECT SYN slightly delayed?

– If SYN/ACK with CWR and ECE set and NS =0 (AccECN supported and no CE in SYN), the client continues with cwdn=W0 and uses AccECN

– If SYN/ACK with CWR, NS and ECE set (AccECN supported and CE in SYN), the client continues with cwnd=1 SMSS and uses AccECN

– If SYN/ACK with ECE set and CWR reset (AccECN/this spec not supported, RFC3168 supported), the client continues using RFC3168. ***What cwnd to use? W0? 1? W0/2?***

– If SYN/ACK with any other combination (RFC3168 not supported), client continues without ECN. ***What cwnd to use? W0? 1? W0/2?***

# Pure ACK

- TCP endpoint MAY set the ECT(0) or the ECT(1) codepoints in a pure ACK
- If the endpoint receives a congestion signal back, it reacts as in any other packet
  - If the endpoint is only sending pure ACKs, it wont be able to reduce the load by doing this.
  - Shall we explore other means to reduce the load?
    - E.g. Increase the number of ACKs until sending a delayed ACK?
- TCP sender process ECT(0/1) and CE marked pure ACKs as any other packet.

# Window probe

- TCP endpoint MAY set ECT(0) /ECT(1) in a zero window probe (ZWP) packet
- If the sender receives a congestion signal, it will reduce its cwnd accordingly.
- However if it is still with RCVWND=0, then there not much it can do, maybe increasing the ZWP sending interval?
- TCP sender process ECT(0/1) and CE marked ZWP as any other packet.

# RST, FIN and retransmissions

- FINs and RTXs, may be marked, processing as any other pkt
- RSTs, hardly useful as a congestion notification vehicle, since there is connection at the sender. The only motivation is to void dropping them more frequently.
  - for senders, stacks MUST allow for administrators to configure whether the RST messages are marked with the ECT(0) or ECT(1) codepoints. We should define a default behaviour, not sure which that one should be.
  - for receivers, ECT and CE codepoints are ignored.

# Next steps?

- From the authors perspective, the document is ready to call for adoption.