Problem Statement and Requirements for a More Accurate ECN Feedback

tcpm – 85. IETF Atlanta – Nov 6, 2012

draft-kuehlewind-tcpm-accecn-reqs-00

Mirja Kühlewind <mirja.kuehlewind@ikr.uni-stuttgart.de>
Richard Scheffenegger <rs@netapp.com>

Problem Statement

Explicit Congestion Notification (ECN)

- allows marking packets instead of dropping in case of congestion
- but provides only one congestion feedback signal per RTT and
- does not announce the total number of marking to the sender

→ New TCP mechanisms need to know how many congestion markings occurred (ConEx, DCTCP and potentially other congestion control algorithms)

→ Standardize a new ECN feedback mechanism within TCP that continually feeds back the extent of congestion, not merely its existence

Requirements

- Resilience

  Take delayed ACK and ACK loss into account (also in situations of high congestion)

- Timely feedback

  Deliver within one RTT (plus additional delays by delayed ACKs)

- Integrity

  Detect misbehaving receiver or network node (as least as good as ECN Nonce)

- Accuracy (+ reliability)

  Ensure to receive at least one congestion notification per RTT (as classic ECN)

  → A sender must not assume to get the exact number of congestion marking in all situations

- Complexity

  Implementation should be as simple as possible and only a minimum of addition state information should be needed

- Network load

  Limit additional network load (when using additional header space or more frequent ACKs)

- Middlebox traversal

  Provide a fallback in case of middelboxes dropping packets with new ECN feedback

Design Approaches

- Re-use of ECN/Nonce (ECE, CWR, NS) Header Bits
  - For capacity negotiation in TCP handshake (*draft-briscoe-conex-re-ecn-tcp*)
  - 1 bit scheme = send ECE once for every CE received (*DCTCP* and *draft-kuehlewind-tcpm-accurate-ecn-00)*
  - 3 bit CE counter (*draft-briscoe-conex-re-ecn-tcp*)
  - codepoint scheme (*draft-kuehlewind-tcpm-accurate-ecn-01*)
- Re-use of other Header Bits

  2 bit counter scheme plus additional bits of the TCP Urgent Pointer field if not needed otherwise (*Bob Briscoe*)
- Use of Reserved Bits
  - Use of above proposed schemes in addition to the classic ECN (reliable feedback per RTT)
  - Extend schemes above to improve robustness against ACK lost
- TCP Option
  - In addition to classic ECN or one of the proposed schemes (*draft-kuehlewind-tcpm-accurate-ecn-option*)
  - Additional option space can be used to provide further information as exact number of marker/lost bytes

1 Bit Scheme

- Send one ECE for each CE received (use CWR in subsequent ACK to increase redundancy)
- Use delayed ACK only if CE status does not change, otherwise send ACK immediately

Discussion

- ACK loss
  - Loss of two subsequent ACKs could result in complete loss of the congestion information
  - Proposed immediate ACK scheme can increase ACK (in worst case to one ACK per data packet)
- ECN Nonce

  NS bit is not used otherwise

Pro: Low complexity and ECN Nonce integrity check supported

Contra: Low robustness against ACK loss

2/3 Bit Counter

Use ECE, CWR (and NS) to send least significant bit of CE counter in every ACK

Discussion

- ACK loss
  - 3 bit counter provides robustness against 4 subsequence ACK losses with delayed ACKs
  - Use of additional header bits (e.g. Urgend Pointer field) can improve robustness
- ECN Nonce

  3 bit counter does use the NS but does not implement any other integrity check

Pro: Quite low complexity

Contra: No integrity check

3 Bit Codepoint Scheme

- Use ECE, CWR, and NS bit to encode 8 codepoint (5 for CE counter and 3 for ECT(1) counter as ECN Nonce)
- See https://datatracker.ietf.org/ipr/1881/

Discussion

- ACK loss
  - Up-to two consecutive ACKs with 100% CE marking rate can be tolerated
  - At low congestion higher numbers of consecutive ACKs may be lost
- ECN Nonce

  Provides more accurate information than ECN Nonce

Pro: Resiliency and integrity

Contra: Complexity

TCP Option

- Negotitation in TCP handshake with an abbreviated option
- 1 or 2 byte counter of ECT(0), ECT(1), CE, non-ECT, and lost packets plus total bytes of CE marked packets

→ Always in addition to ECE, CWR, and NS bits in TCP header (no matter if used for classic ECN or a new ECN feedback scheme)

> **Note:** Using Classic ECN in addition can provide at least one congestion feedback signal per RTT reliably

Pro: High accuracy also for integrity check

Contra: Additional header space need in all (?) packets, problem with middelboxes?