TCP modifications for Congestion Exposure

ConEx – 82. IETF Taipei – November 17, 2011

draft-kuehlewind-conex-tcp-modifications-01

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

Sender-side Modifications

A ConEx sender MUST ...

- "... expose congestion to the network according to the congestion information received by ECN or based on loss provided by the TCP feedback loop."
  → Only sender-side modifications needed
  → Both half-connections of one TCP connection can enable ConEx independently

- "... negotiate for both SACK (SACK-Permitted Option in SYN, RFC 2018) and ECN or the more accurate ECN feedback in the TCP handshake **if these TCP extension are available at the sender.**"
  → It's not required to implement more accurate ECN feedback but if available it must be used

Timeliness of the ConEx Signals

- The sender SHOULD sent the ConEx signaling with the next available packet
- The sender MUST NOT delay the ConEx signal more than one RTT

Accounting Congestion

Byte-wise Accounting

From draft-ietf-conex-abstract-mech:

> "Within any flow or aggregate of flows, the volume of data (**total number of bytes**) tagged with ConEx Signals should never be less than the total volume of ECN marked data seen near the receiver."

- A TCP ConEx sender MUST account congestion byte-wise (and not packet-wise)
- A ConEx sender MAY only account the TCP payload bytes (if packets are equal sized)

  → The ConEx marked packets as well as the original packets causing the congestion will both contain about the same number of headers
- Otherwise the sender MUST take the headers into account
- A ConEx sender MUST mark the respective number of payload bytes in subsequent packets (after the congestion notification)

Accounting Congestion

ECN-based Congestion feedback

Congestion Exposure Gauge (CEG): num. of outstanding bytes with E bit

Accurate ECN feedback

→ **`CEG += min( SMSS*D, acked_bytes )`**
D is the number of ECN feedback marks (calculation depends on the coding)

Classic ECN support

1. Full compliance mode (Only one ECN feedback signal per RTT)
   → **`CEG += SMSS`** (whenever the ECE flag toggles from "0" to "1")
2. Simple compatibility mode
   – Set the CWR permanently to force the receiver to signal only one ECE per CE mark
   – Problem with delayed ACKs will cause information loss in high congestion situation
   – Proposed solution: Assume every received marking as M markings (M=2 delayed ACKs)
   → **`CEG += M*SMSS`** (for every ECE flag)
3. Advanced compatibility mode
   – Set CWR only on those data segments, that will actually trigger an (delayed) ACK
   → if previous_marked: **`CEG += min( M*SSMS, acked_bytes )`**, else: **`CEG += SMSS`**

Accounting Congestion

Loss Detection with/without SACK

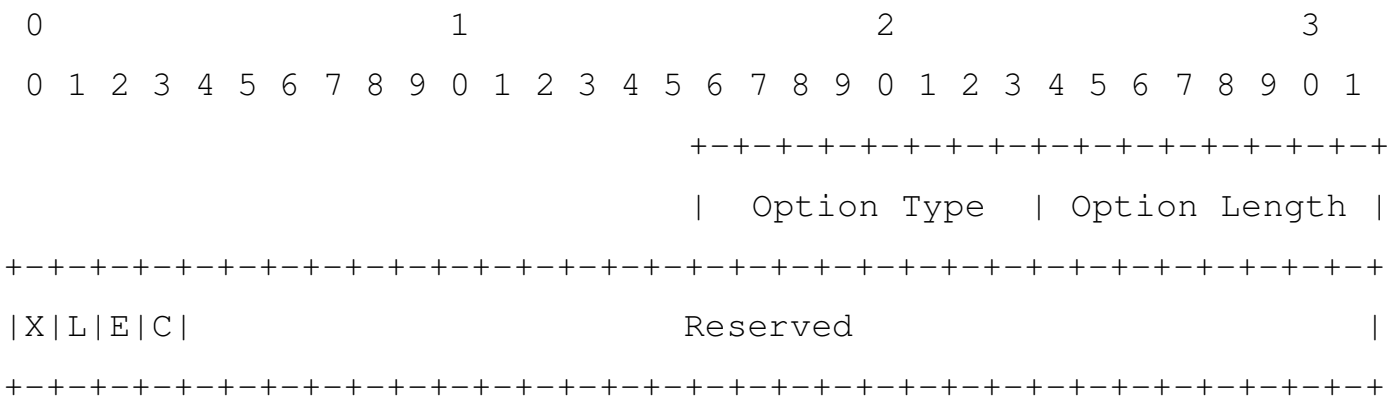
Loss Exposure Gauge (LEG): number of outstanding bytes with L bit

- Increase LEG by the size of the TCP payload containing a retransmission (if equal sized packets are sent)

  → L bit is set on subsequent packet
- Decrease LEG if spurious retransmit have been detected

  → LEG can get negative

Setting the ConEx IPv6 Bits



Setting the X bit

- All packets carrying payload MUST be marked with the X bit set (including retransmissions)
- Control packets as pure ACKs (which are not carrying any payload) MUST carry a ConEx Destination Option with the X bit unset

  → No congestion feedback information is available about those packets

  → Should not be taken into account when determining ConEx information

Setting the ConEx IPv6 Bits

Setting the E Bit and the L Bit

- As long as the CEG/LEG is positive, ConEx-capable packets MUST be marked with E or respective L and the CEG/LEG is decreased by the TCP payload bytes carried in this packet
- If the CEG/LEG is negative, the CEG/LEG is drained by one byte with every packet sent out, as ConEx information are only meaningful for a certain time

→ if `CEG > 0:` **`CEG -= TCPpayload.length`**, else: **`CEG++`**

→ if `LEG > 0:` **`LEG -= TCPpayload.length`**, else: **`LEG++`**

Setting the ConEx IPv6 Bits

Setting C(redit) Bits

From draft-ietf-conex-abstract-mech:

"The transport SHOULD signal sufficient credit in advance to cover any reasonably expected congestion during its feedback delay."

→ Credits should cover the increase of CWND per RTT (as this can cause congestion)

Slow Start (RFC5681 congestion control)

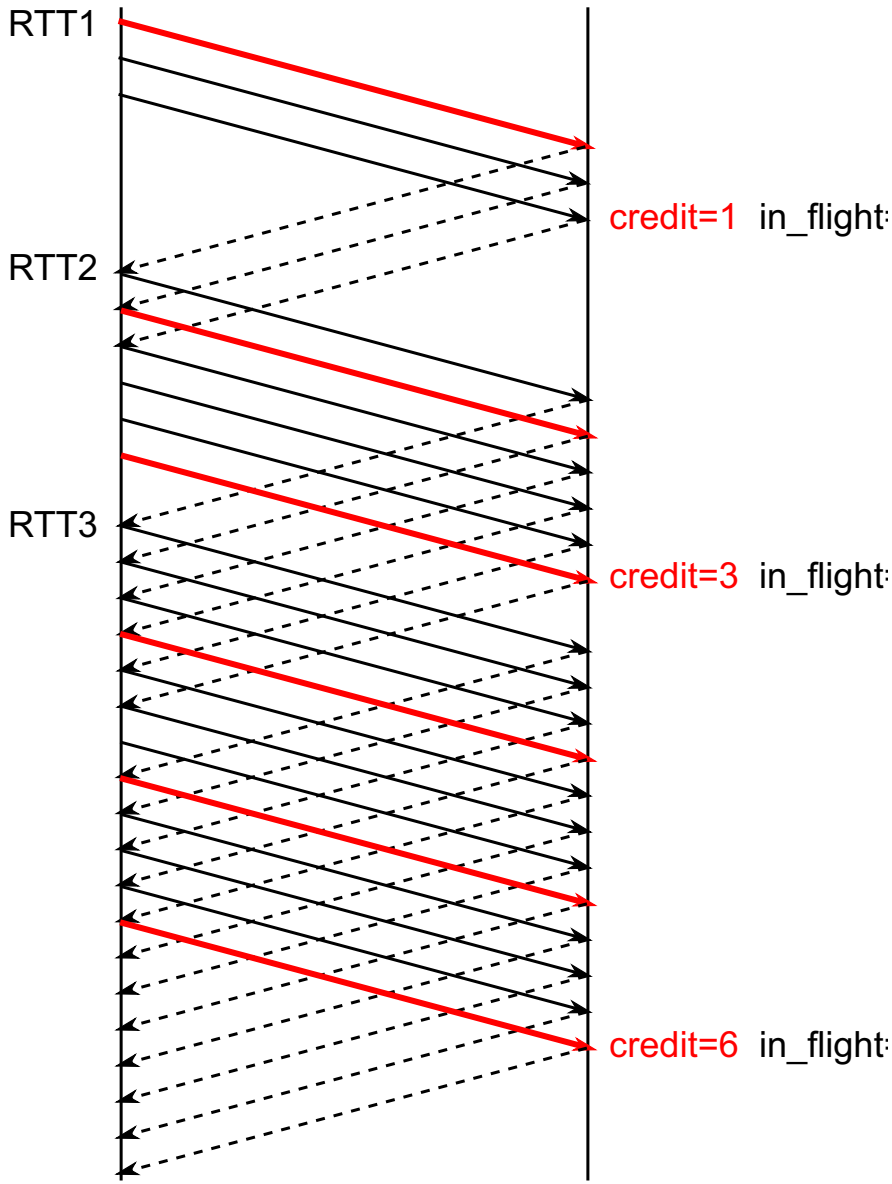
Exponential increase means double CWND very RTT

→ Half of the flight size has to be marked

→ Marking of every fourth packet (as credit will not time out during Slow Start phase)

Increasing number of losses

→ Can indicate losses incorporated by audit device → Sender should send further credits



Question?