TCP Modifications for Congestion Exposure

ConEx – 87. IETF Berlin – July 27, 2013

draft-ietf-conex-tcp-modifications-04

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

Updates -03 and -04

Credit handling

- In Slow Start: mark every 4. packet
- In Congestion Avoidance: often no further credits are needed
  - count number of sent credits in counter c
  - monitor number of packets in flight f

  → if f > c, send new credits
- Loss of ConEx-marked packets: detect and send further credits

  → if losses occur in two subsequent RTTs, reset the credit count c (reactive)

→ Needs to be changed, if credit definition changes!

Classic ECN full compliance mode

Increase Congestion Exposure Gauge (CEG) when ECE flag triggers from 0 to 1

CEG += min(SMSS, DeliveredData)

→ Underestimates the number ECN-(CE)-marks and might case sanctions by an audit

→ Credits of Slow Start will cover mismatch for short connections with only light congestion

→ Otherwise increase CEG (by DeliveredData) for each ACK with ECE bit set

Review comments by Jana

- 2: Sender-side Modifications: "MUST negotiate for both SACK and ECN or the more accurate ECN feedback ..." : *This strikes me as an odd MUST. SHOULD seems adequate.*

  → MUST to support ECN and SACK deployment and make ConEx information most valuable
- "A ConEx sender MUST expose congestion to the network...": *A compliant Conex sender has to follow a Conex spec for exposing congestion; that can be assumed here, without having a MUST in this document.*

  → Change to "A ConEx sender MUST expose **all** congestion information..."
- 3.1.2: Classic ECN Support: *It is non-trivial for a sender to determine when delayed acks will be sent by the receiver, in particular with bidirectional data transfer. I would be careful about suggesting such heuristics without getting into details. Is this "Advanced Compatibility" really practical or necessary?*

  → Describe this option, as ConEx with 'classic' ECN is hardly usable...
- 3.2: Loss detection with/without SACK: "assuming equal sized segments such that the retransmitted packet will have the same number of header as the original ones." *You cannot make this assumption. [...] I would suggest dropping it from the text.*

  → Only a detailed solution for equal sized packets described

Summary

→ No further open issues (if credit definition does not change)

→ Reviews needed!

→ Ready for WGLC (if credit definition does not change)

Backup

TCP modifications for Congestion Exposure

Sender-side Modifications

A ConEx sender MUST negotiate for both SACK (SACK-Permitted Option in SYN, RFC 2018) and the more accurate ECN feedback in the TCP handshake

Setting the ConEx IPv6 Bits

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
                                +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                                |  Option Type  | Option Length |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|X|L|E|C|                     Reserved                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- Setting the X bit

  → **Which packets should be ConEx-capable?** Control pkts/pure ACKs and/or retransmits...
- Byte-wise accounting of the ConEx markings (L, E, C)

  → **Should packets be accounted by their respective IP packet size?**

TCP modifications for Congestion Exposure

Setting the E Bit

Accurate ECN feedback

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

On ACK: CEG += min(SMSS*D, DeliveredData)

D is the number of ECN feedback marks (calculation depends on the coding)

DeliveredData = acked_bytes + SACK_diff + (is_dup)*1SMSS -
(is_after_dup)*num_dup*1SMSS

Classic ECN support

1. Full compliance mode

   Only one ECN feedback signal per RTT

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)

3. Advanced compatibility mode

   More sophisticated scheme to set CWR in the right packets to avoid information loss

TCP modifications for Congestion Exposure

Setting the L Bit: Loss Detection with/without SACK

- **Loss Exposure Gauge (LEG)**: number of outstanding bytes with L bit
  1. Increase LEG by the size of the IP packet containing a retransmission
  2. L bit is set on subsequent packet; LEG is decreased by the size of the sent IP pkt

  → This decouples the ConEx mark from the retransmissions themselves, but also delays it...
- Decrease LEG if spurious retransmit have been detected

  LEG can get negative but should be drained slow as congestion information might time out

TCP modifications for Congestion Exposure

Setting C(redit) Bits

```
"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

Exponential inc. doubles CWND per RTT

→ Halve the flight size has to be marked

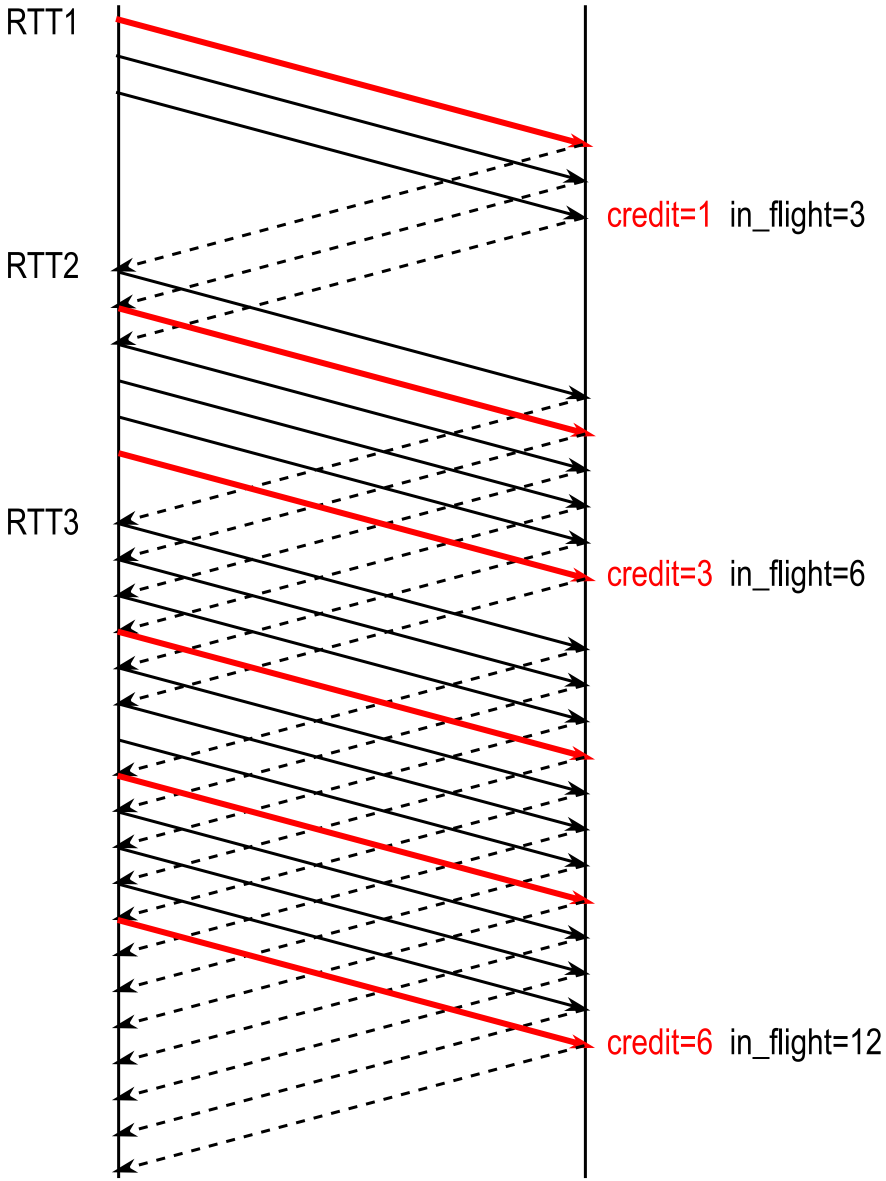
→ Marking of every fourth packet (as credit will not time out during Slow Start phase)

Congestion Avoidance

If fightsize f > credit count c, send new credits

Loss of ConEx-marked packets

Detect and send further credits (reset c)


RTT1
RTT2
RTT3
credit=1 in_flight=3
credit=3 in_flight=6
credit=6 in_flight=12