TCP modifications for Congestion Exposure

ConEx – 81. IETF Quebec – July 27, 2011

draft-kuehlewind-conex-accurate-ecn-00
draft-kuehlewind-conex-tcp-modifications-00

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

New Drafts

→ TCP modifications have been splitted up into two draft

1. Accurate ECN Feedback in TCP

   (draft-kuehlewind-conex-accurate-ecn-00)

   – Mechanism to retrieve more accurate ECN feedback (more than one signal per RTT)
   – Can also be used by other TCP mechanisms. e.g. DCTCP; not ConEx specific
   – Currently 3 different coding scheme proposed and discussed
   → The goal is to chose one of the scheme (remove the other option form the draft) and specify the protocol

2. TCP modifications for Congestion Exposure

   (draft-kuehlewind-conex-tcp-modifications-00)

   – Modification and recommendation for a sender to use ConEx in TCP
   – e.g. use of SACK and accurate ECN feedback, counting congestion signals, handling credits
   → Several open points; more discussion needed

Accurate ECN Feedback in TCP

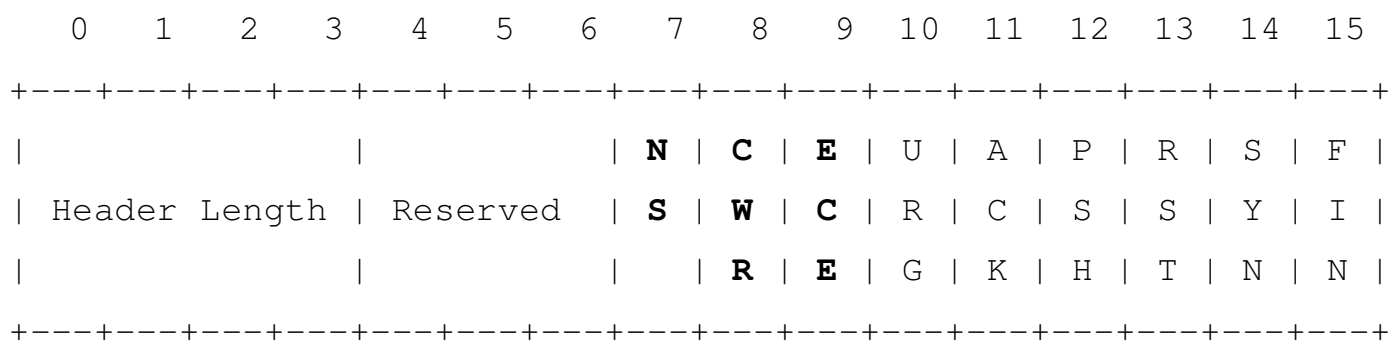
Overview ECN and ECN Nonce in TCP

Terminology from [RFC3168] and [RFC3540]

The ECN field in the IP header

- ECT(0)/ECT(1): either one of the two ECN-Capable Transport codepoints
- CE: the Congestion Experienced codepoint

The ECN flags in bytes 13 and 14 of the TCP Header



- CWR: the Congestion Window Reduced flag
- ECE: the ECN-Echo flag
- NS: ECN Nonce Sum

Accurate ECN Feedback in TCP

Design Choices

- Re-use of the ECN/ECN-Nonce TCP bits

  Classic ECN should not be used in parallel anymore
- No additional bits from three reserved bits in TCP header

  No additional benefit (only shift of problems in time)
- No extra TCP Option
  - Deployment issues because of middleboxes
  - Growth of header length (goal would be to have this mechanism activated by default)
  - Could provides more information e.g. explicit the number of ECT(0), ECT(1), CE, non- ECT marked and lost packets (as in ECN for RTP/UDP), but is this needed?

Accurate ECN Feedback in TCP

Negotiation in the TCP Handshake

1. Host A indicates a request to get more accurate ECN feedback by setting **NS=1, CWR=1** and **ECE=1** in the **initial SYN**

   Classic ECN will still be negotiated (with CWR=1 and ECE=1)

2. Host B returns a **SYN ACK** with flags **CWR=1** and **ECE=0**

   Broken receiver that just reflect SYN bits get detected

Ac	N	E	I	[SYN] A->B	[SYN,ACK] B->A	Mode
				NS CWR ECE	NS CWR ECE	
AB				1 1 1	X 1 0	accurate ECN
A	B			1 1 1	1 0 1	ECN Nonce
A		B		1 1 1	0 0 1	classic ECN
A			B	1 1 1	0 0 0	Not ECN
A			B	1 1 1	1 1 1	Not ECN (broken)

Ac: *Ac*curate ECN Feedback, N: ECN-*N*once (RFC3540), E: *E*CN (RFC3168),
I: Not-ECN (*I*mplicit congestion notification).

Accurate ECN Feedback in TCP

Proposed Accurate Feedback Coding Schemes

- Requirements on resilience, timeliness, integrity, accuracy and complexity listed
- Discussion (ACK loss, ECN Nonce) not exhausting yet...

  → Please read draft and mention all possible pros and cons on the list!

Three coding options proposed

1. One bit feedback flag
   - Signal ECE only in one (N subsequent) ACKs
   - Remark: In one ACK all acknowledged bytes are regarded as congested (not in draft...)
   - Remark: CWR is unused; can be used for redundancy in subsequent ACK (not in draft...)
2. Three bit field with counter feedback
   - Use ECE/CWR/NS signal a counter value (mod8) in every ACK (as with re-ECN)
   - Does not allow ECN Nonce
3. Codepoints with dual counter feedback

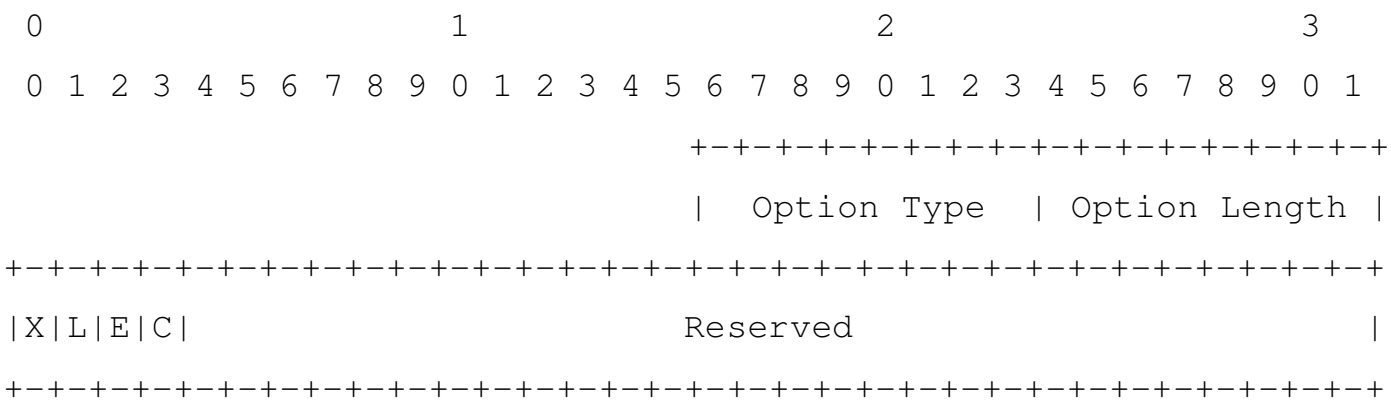
   Have 2 counter (CE, ECT(1)) encoded in 8 codepoints (send congestion value by default)

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



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

CEG += min((SMSS+IP.header+TCP.header)*D, acked_bytes + (IP+TCP Header)*D)

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

→ Document all three schemes as choice might depend on sender capabilities

→ Does this belong here or in the other doc?

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 (RFC5681 congestion control)

Exponential increase means double CWND very RTT

→ Halve the flight size has to be marked

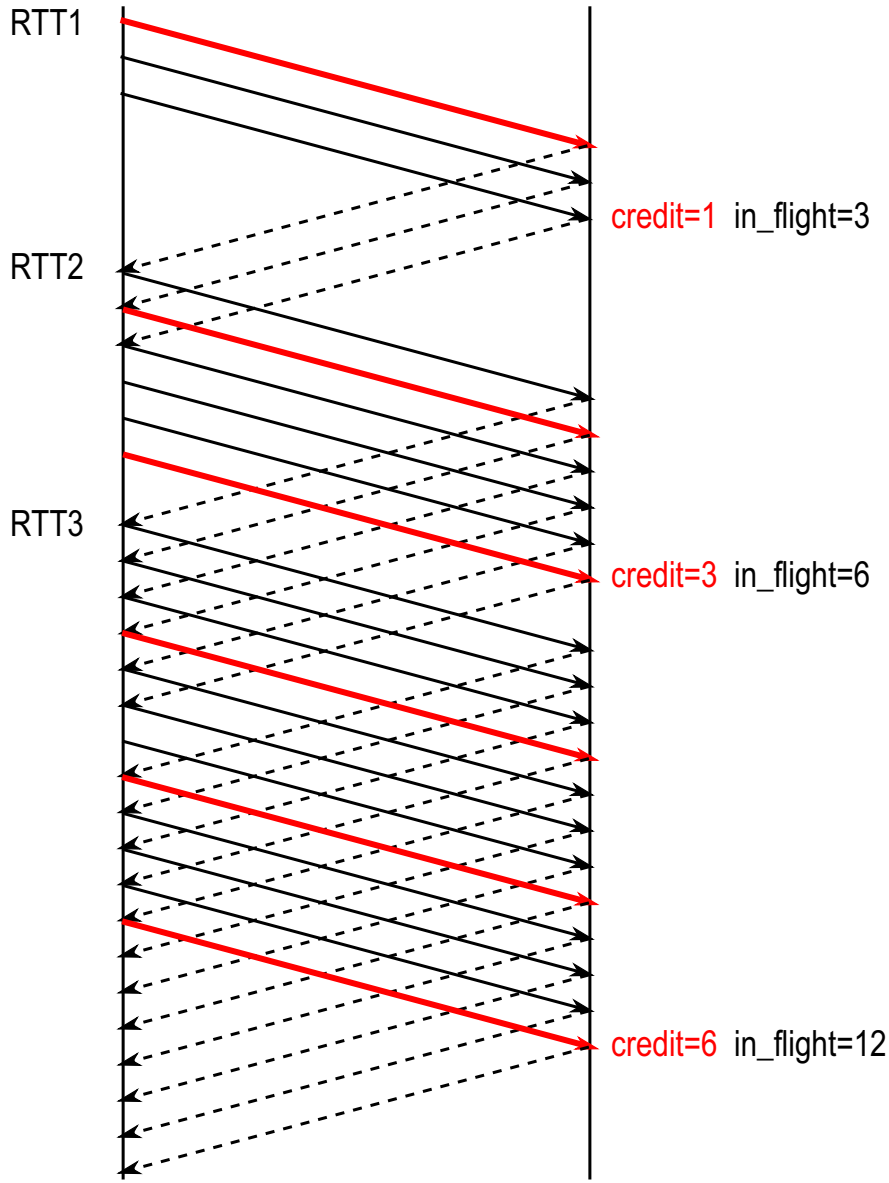
→ Marking of every fourth packet (as credit will not time out during Slow Start phase)

Increasing number of losses

can indicate losses incorporated by audit device

→ Sender should send further credits

→ Expiration of credits?



TCP modifications for Congestion Exposure

Timeliness of the ConEx Signals

Recommendations

- Sender should not delay ConEx signaling excessively
- Space out of the signaling of multiple markings across a (short) period of time (within one RTT) is possible
- Marking of retransmission is possible

Open Issues

- Marking of control packets? (Byte-wise accounting: only possible if IP packet size is regarded)
- Expiration of the ConEx information? (credits, echoed congestion)
- Further recommendations on congestion control needed? (e.g different crediting when restarting a transmission on a known link)

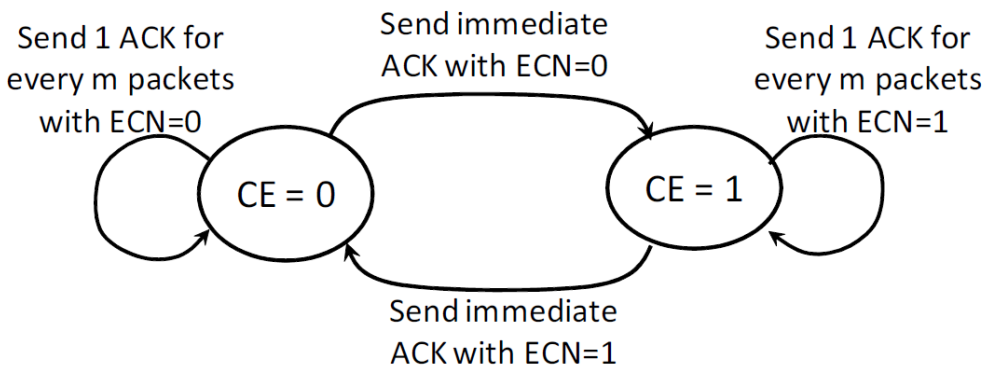
Question?

Backup

Accurate ECN Feedback in TCP

One Bit Feedback Flag

- Set ECE bit in only one ACK when CE is received
  → No secured transmission; ACK might get lost
- Possiblity to repeat the same ACK N(=2) times
  → Delays all feedback information, even worse with delayed ACKs
- Immediately send ACK if congestion situation changes



Remark: In one Acknowledgment all acknowledged bytes are regarded as congested

Discussion

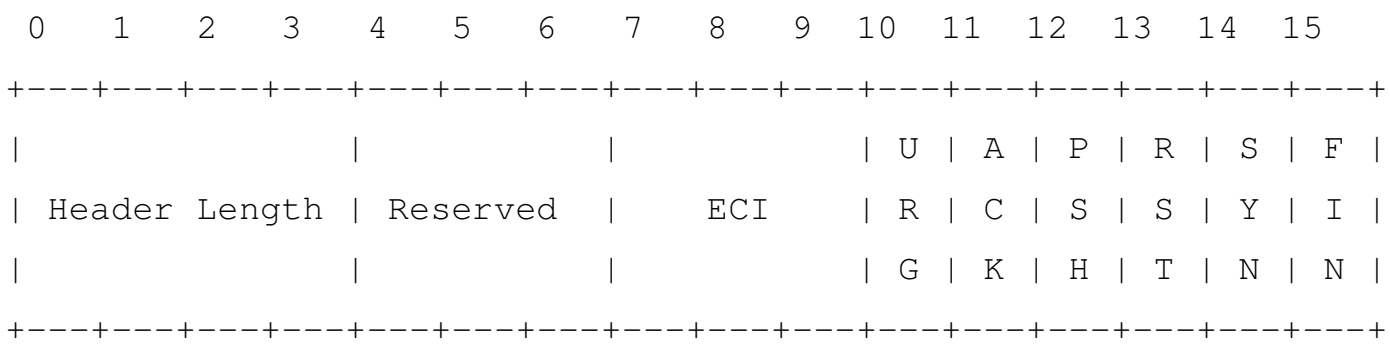
- ACK loss
- ECN Nonce can still be used in parallel

Accurate ECN Feedback in TCP

Three Bit Field with Counter Feedback

Echo Congestion Counter (ECC): number of CE marked packet during a half-connection

Echo Congestion Increment (ECI): 3-bit field for the receiver to permanently signal the sender the current value of ECC, modulo 8, with each ACK



Accurate ECN Feedback in TCP

Codepoints with Dual Counter Feedback

One field in TCP ACK but encoding 2 counters in 8 codepoints

1. Congestion Indication (CI) counter: number of CE marks
2. ECT(1) (E1) counter: number of ECT(1) signals

ECI	NS	CWR	ECE	CI (base5)	E1 (base3)
0	0	0	0	0	-
1	0	0	1	1	-
2	0	1	0	2	-
3	0	1	1	3	-
4	1	0	0	4	-
5	1	0	1	-	0
6	1	1	0	-	1
7	1	1	1	-	2

– By default an accurate ECN receiver MUST echo the CI counter (modulo 5)
– The receiver MUST repeat the codepoint directly on the subsequent ACK
– Whenever ECT(1) occurs, E1 will be echoed (twice); expect CE is observed at same time

Accurate ECN Feedback in TCP

Discussion

Section	Resiliency	Timely	Integrity	Accuracy	Complexity
1-bit-flag	-	+	+	-	+
3-bit-field	++	++	--	++	-
Codepoints	+	+	+	++	--

Which should we take?