Accurate ECN Feedback in TCP

TCPM– 82. IETF Taipei – November 16, 2011

draft-kuehlewind-conex-accurate-ecn-01

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

Drafts

1. Accurate ECN Feedback in TCP

   (draft-kuehlewind-conex-accurate-ecn-01)

   - Mechanism to retrieve more accurate ECN feedback (more than one signal per RTT) by reusing the 3 ECN/ECN-Nonce bits in the TCP header
   - 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. Accurate ECN Feedback Option in TCP

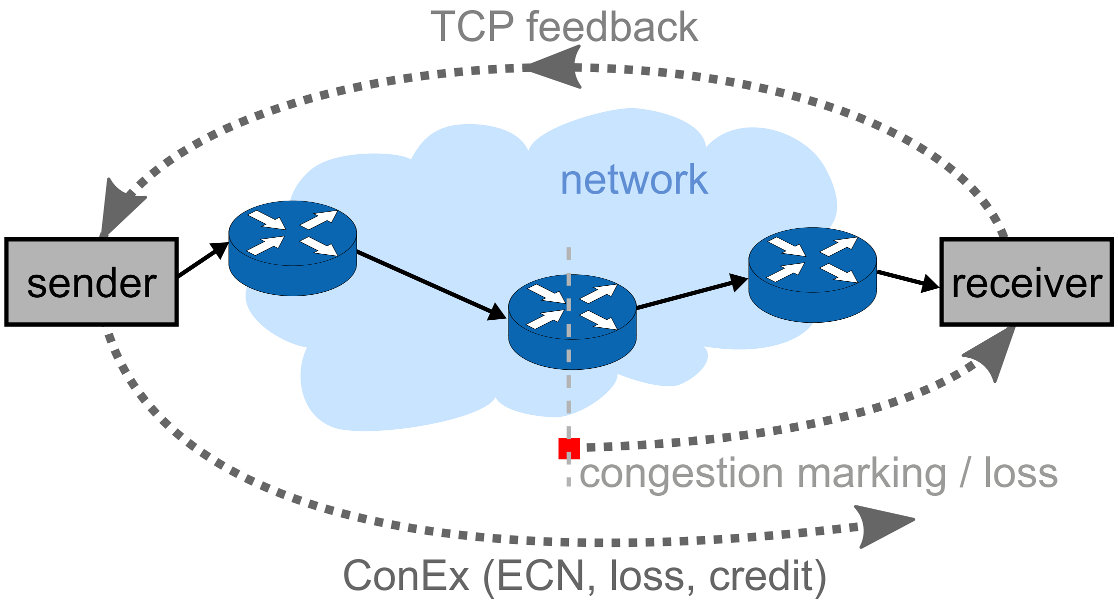
   (draft-kuehlewind-tcpm-accurate-ecn-option-00)

   - TCP Option for accurate ECN feedback
   - Can be used in addition to the classic ECN or the scheme above
   - Not proposed as default mechanism because of
     - Middlebox issues with new TCP Options
     - Overhead in TCP header

Congestion Exposure (ConEx)

Congestion Exposure Principle

Mechanism by which senders inform the network about the congestion encountered by previous packets on the same flow, either loss or ECN markings



→ Re-insert congestion information (= estimation about expected congestion level)

How to use ConEx?

- Reveal rest-of-path congestion along the path (to any intermediate node)
- Make senders accountable for congestion they cause by network ingrees policing

Accurate ECN Feedback in TCP

Problem

ECN provides only one congestion feedback signal per RTT ...

as designed for current congestion control mechanisms
that will react only once per RTT on congestion

... but ConEx needs to know how many congestion markings occured exactly.

Scope

- Congestion control might react differently on ECN in future (ICCRG and LEDBAT ML)
- Can also be used by other TCP mechanisms, e.g. DCTCP

→ Not ConEx specific

→ Feedback of tcpm community 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

```
  0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|               |           | N | C | E | U | A | P | R | S | F |
| Header Length | Reserved  | S | W | C | R | C | S | S | Y | I |
|               |           |   | R | E | G | K | H | T | N | N |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

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

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

Three coding options proposed

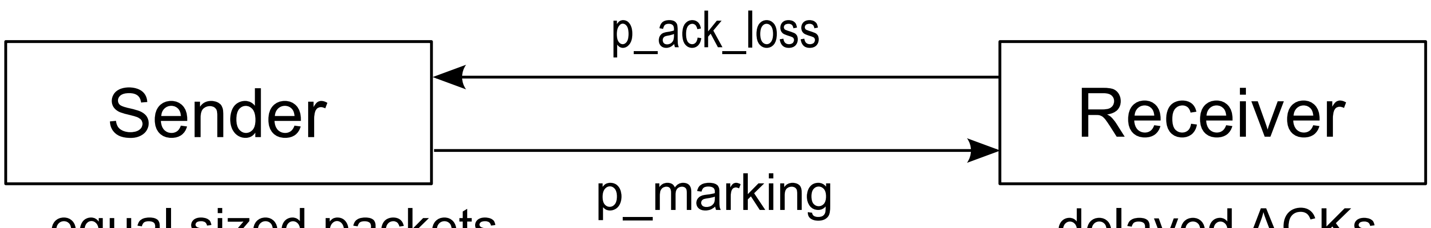
1. One bit feedback flag
   - Signal ECE only in one ACKs
   - Set CWR for redundancy in subsequent ACK: `CWR(t) = ECE(t-1)`
   - Immediate ACK'ing when CE changes (instead of delayed ACKs)
   - Byte-wise: In one ACK all acknowledged bytes are regarded as congestion marked
2. Three bit field with counter feedback
   - Use ECE/CWR/NS signal a counter value (mod8) of number of CE marks in every ACK
   - Does not allow ECN Nonce
3. Codepoints with dual counter feedback
   - Have 2 counter (CI for congestion indications, E1 for ECT(1)) encoded in 8 codepoints
   - Send value of CI counter by default; send E1 counter value in next ACK if ECT(1) received

→ Discussion (ACK loss, ECN Nonce) not exhausting yet...

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

Simulation

Simple Scenario



Sender

Receiver

p_ack_loss

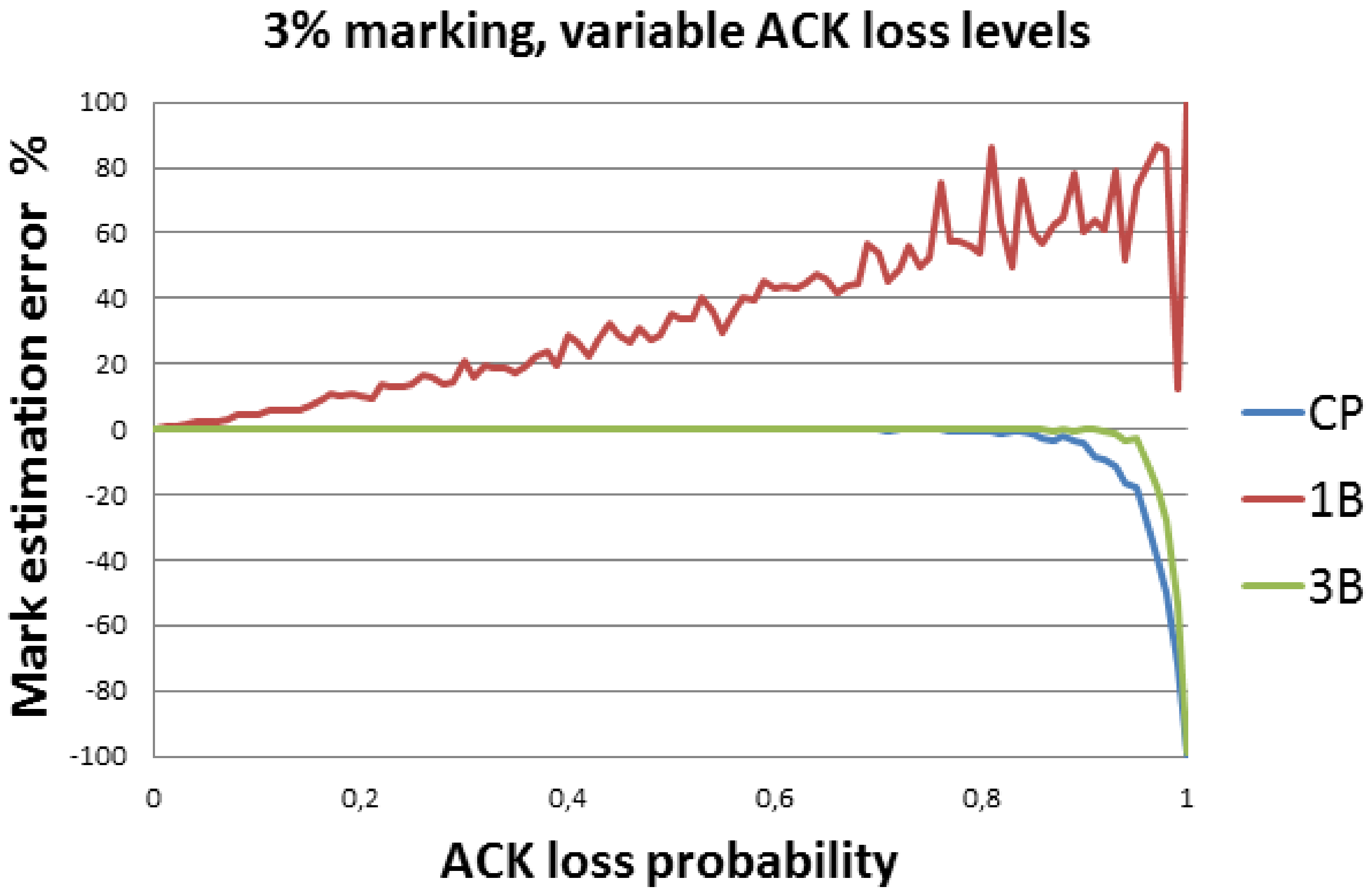
p_marking

- equal sized packets
- Num_est

- delayed ACKs
- Num_CE

Simulation

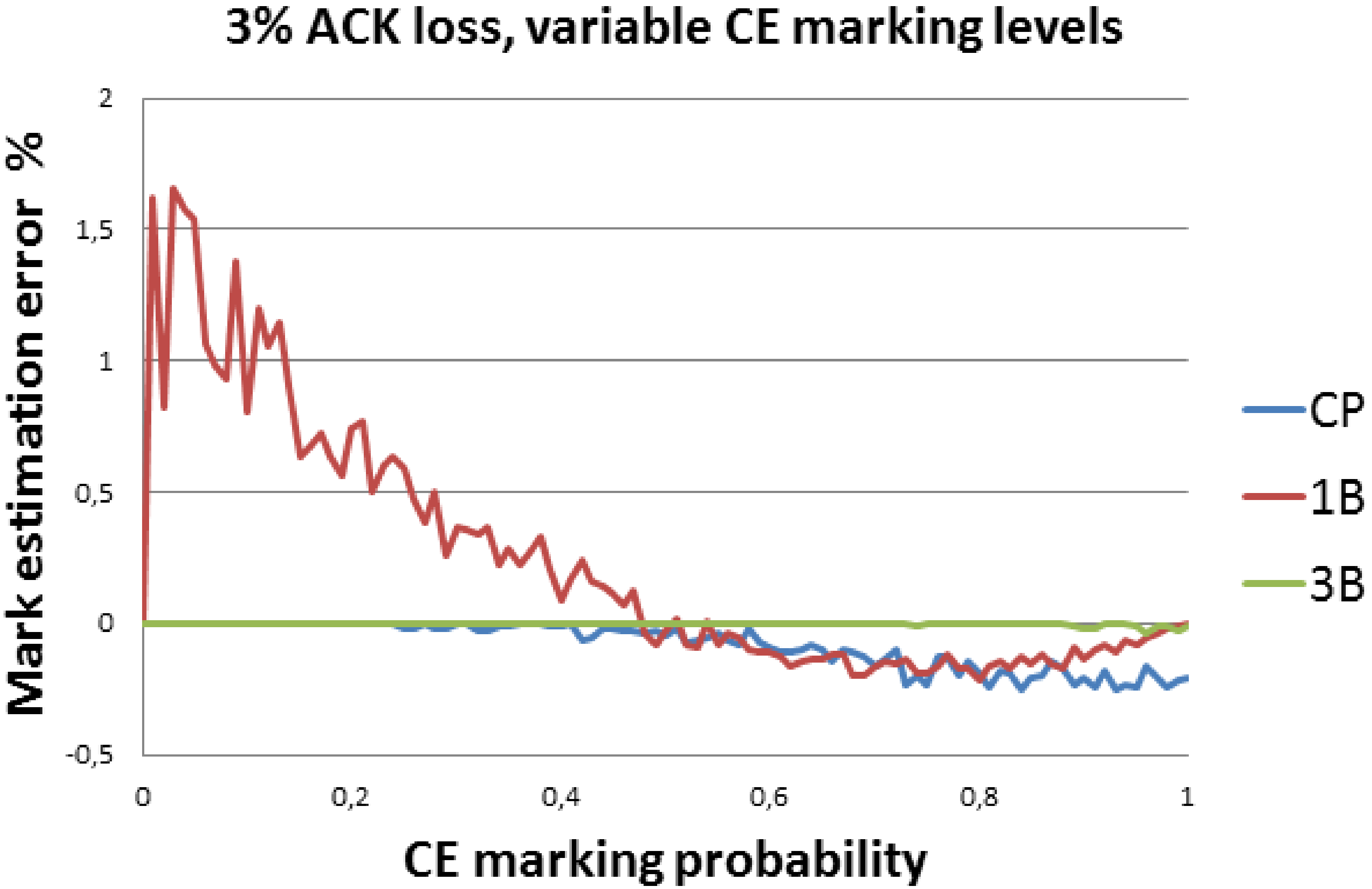
Preliminary Results



→ Is underestimating worse than overestimating as a single CE in a RTT can be missed?

Simulation

Preliminary Results



Simulation

Current Simulation Setup

Limitations

- Packet counting instead of byte counting (as equal sized packets assumed)

  → 1 Bit variant is only one that is taking the number of ack'ed bytes into account
- Gaussian distributed marks/ACK losses

  → Accumulated marks/losses (in one RTT) in reality might give different results
- Total % of over-/underestimation is given

  → Look at % of missed congestion events (= all markings within one RTT)

Conclusion

- 3 Bit (3B) variant gives upper bound but ...

  no ECN-Nonce support
- Codepoint (CP) variant seems to be better than 1 Bit (1B) variant but ...
  - more complex (?)
  - no byte-wise accounting
- In 1 Bit (1B) variant also the number of ACKs increases

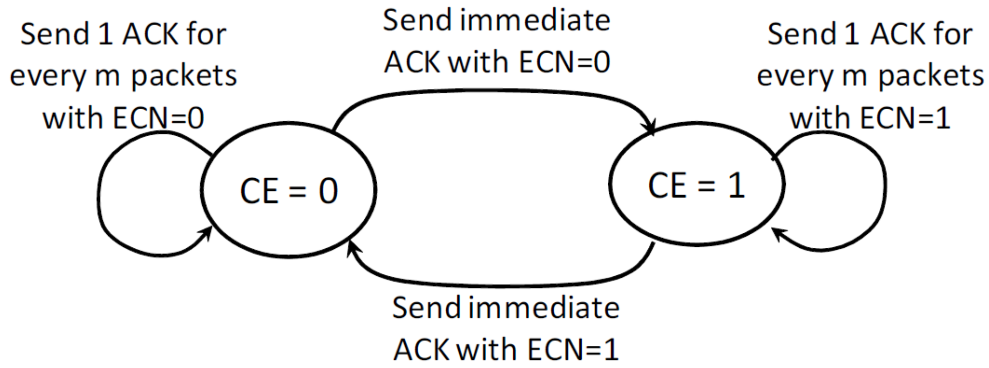
Question?

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 (proposed by DCTCP, Microsoft)


Send 1 ACK for every m packets with ECN=0
Send immediate ACK with ECN=0
Send 1 ACK for every m packets with ECN=1
CE = 0
CE = 1
Send immediate ACK with ECN=1

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

```
  0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|               |           |           | U | A | P | R | S | F |
| Header Length | Reserved  |    ECI    | R | C | S | S | Y | I |
|               |           |           | G | K | H | T | N | N |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

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; 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	+	+	+	++	--