

# More Accurate ECN Feedback in TCP

draft-kuehlewind-tcpm-accurate-ecn-05

Bob Briscoe <[ietf@bobbriscoe.net](mailto:ietf@bobbriscoe.net)>

**Mirja Kühlewind** <[mirja.kuehlewind@tik.ee.ethz.ch](mailto:mirja.kuehlewind@tik.ee.ethz.ch)>

Richard Scheffenegger <[rscheff@gmx.at](mailto:rscheff@gmx.at)>

# Background & Problem

- **Explicit Congestion Notification (ECN):** Routers mark packets as Congestion Experienced (CE) instead of dropping them in case of incipient congestion
- **ECN Feedback in RFC6831:** Receiver only provides feedback once per RTT to the sender
- **Accurate ECN (AccECN):** Receiver feeds back the accurate number of seen markings (within each RTT)

# History

- Oct 2005: First idea in draft-briscoe-tsvwg-re-ecn-tcp
- July 2011: draft-kuehlewind-conex-accurate-ecn-00
- Dec 2011: First version of draft-kuehlewind-tcpm-accurate-ecn
- Aug 2015: RFC7560 - Problem Statement and Requirements for Increased Accuracy in ECN Feedback
- Sep 2015: New re-worked and simpler version of draft-kuehlewind-tcpm-accurate-ecn -> Ready to move on!

# Recent History

- RFC7560 recognized not all reqs might be achievable
  - ➔ compromises would be necessary
- -03: extreme measures for ultra-robust middlebox traversal and zero header overhead
  - ➔ list feedback: too complex
- -04: essential part will traverse mboxes, so removed all the complexity around traversal of optional part
  - ➔ positive feedback from 2 reviews and other more minor comments
- -05: addressed review concerns

# Overview AccECN

- **Capability Negotiation:** Repurposing the NS (ECN Nonce Sum) TCP header flag
    - fully backward compatible
  - **Essential Feedback:** Overloading the ECN TCP header flags (NS/ECE/CWR) as *Accurate ECN (ACE) field*
    - feed back the number of received CE marks (including control packets without payload)
    - no overhead compared to classic ECN but limited resilience to loss
  - **Supplementary Feedback:** Using a new *AccECN TCP Option*
    - provide additional feedback on the number of marked bytes
- **Both essential and supplementary parts:** receiver maintains ECN-IP-codepoint counters and AccECN repeats LSBs of counters for resilience

# AccECN Capability Negotiation

Ac	N	E	I	SYN A->B			SYN/ACK B->A			Feedback Mode
				NS	CWR	ECE	NS	CWR	ECE	
AB				1	1	1	0	1	0	AccECN
AB				1	1	1	1	1	0	AccECN (CE on SYN)
A	B			1	1	1	1	0	1	classic ECN
A		B		1	1	1	0	0	1	classic ECN
A			B	1	1	1	0	0	0	Not ECN
B	A			0	1	1	0	0	1	classic ECN
B		A		0	1	1	0	0	1	classic ECN
B			A	0	0	0	0	0	0	Not ECN
A			B	1	1	1	1	1	1	Not ECN (broken)
A				1	1	1	0	1	1	Not ECN (see Appx B)
A				1	1	1	1	0	0	Not ECN (see Appx B)

**Both ends AccECN-capable**

initiator sends NS/ECE/CWR

responder sends CWR

**Responder not AccECN-capable**

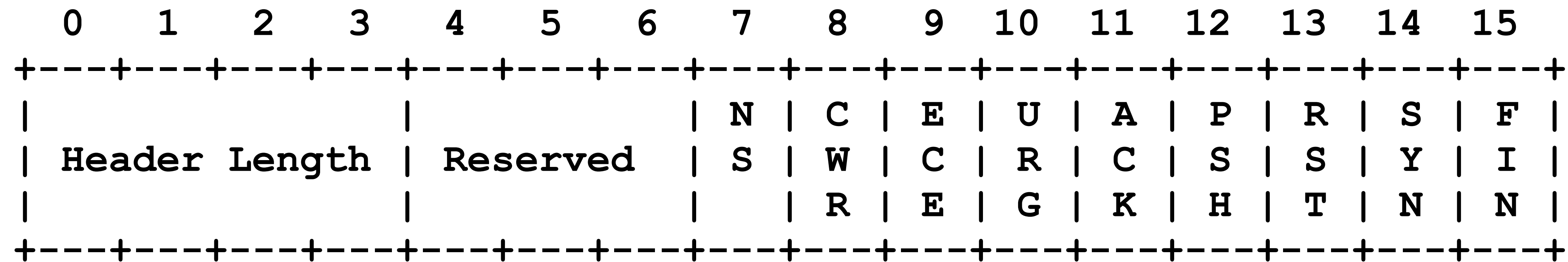
initiator sends NS/ECE/CWR

(ECE/CWR is classic ECN nego)

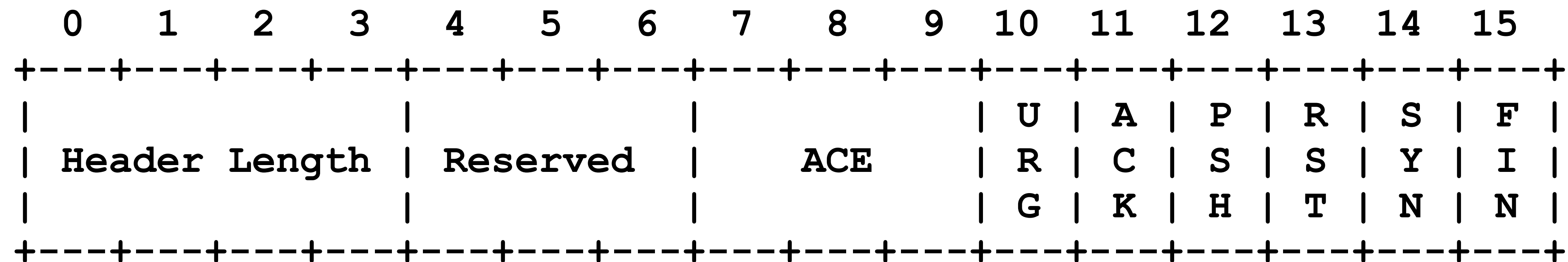
responder sends ECE

# The ACE field

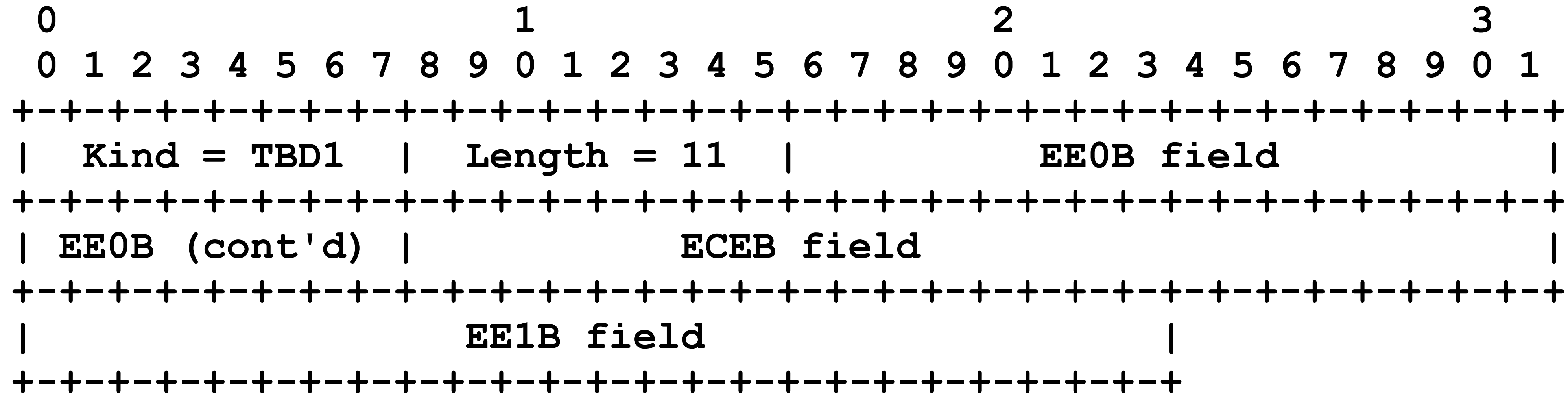
The (post-ECN Nonce) definition of the TCP header flags (bytes 13 & 14):



Definition of the ACE field (when AccECN has been negotiated and SYN=0):



# The AccECN Option



- EE0B** number of bytes received with ECT(0) marked
- ECEB\*** number of bytes received with CE marked
- EE1B\*** number of bytes received with ECT(1) marked

\*optional



# Usage of the AccECN TCP Option

- **Change-Triggered ACKs**  
SHOULD send immediate ACK If an arriving packet increments a different byte counter
- **Continual Repetition**  
SHOULD include if CE-bytes-counter has incremented (MUST give precedence to SACK if space is limited)
- **Full-Length Options Preferred**  
SHOULD always use full-length AccECN Options; MAY use shorter AccECN Options if space is limited, but it MUST include the counter(s) that have incremented since the previous AccECN Option
- **Beaconing Full-Length Options**  
MUST include a full-length AccECN TCP Option on at least three ACKs per RTT

# AccECN Protocol Features Summary

Requirement	Classic ECN	ECN Nonce	DCTCP	AccECN-03			AccECN-04/05	
				Urg-Ptr	TCP opt	essential	TCP opt	essential
Resilience	+	+	-	+	+	o	+	o
Timeliness	o	o	-	+	+	+	+	+
Integrity	-	o	+*	+*	+*	+*	+*	+*
Accuracy	-	-	-	+	+	+	+	+
Ordering	-	-	+	+	+	-	+	-
Simplicity	++	+	o	-	-	o	+	+
Overhead	++	o	o	+	o	++	-	++
Compatibility	o	o	-	o	-	o	-	o

\* = compatible with an independent zero-overhead integrity solution

# Implementation Status

- Linux patch available: <https://github.com/mirjak/linux-accecn/>
  - currently sends Exp AccECN Option with all packets (except SYN)
  - no changed-triggered ACKs yet
  - no counter wrap detection implemented yet
- TCP Experimental Option Experiment Identifier (TCP ExID) registered with IANA:
  - 0xACCE

# Open Issues/Discussions

- Mandatory or optional use of Change-Triggered ACKs?
- See Appendix B & C for alternative design choices and a couple of other minor open issues
- Others?
- Ready for wg adoption?