More Accurate ECN Feedback in TCP

Bob Briscoe, CableLabs®
Mirja Kuhlewind, ETH Zürich
Richard Scheffenegger, NetApp®

TCPM WG, IETF-102, Jul 2018
Problem (Recap): Congestion Existence, not Extent

- Explicit Congestion Notification (ECN)
  - routers switches mark more packets as load grows
  - RFC3168 added ECN to IP and TCP

- Problem with RFC3168 ECN feedback:
  - only one TCP feedback per RTT
  - rcvr repeats ECE flag for reliability, until sender’s CWR flag acks it
  - suited TCP at the time – one congestion response per RTT
Solution (recap): Congestion Extent, not just Existence

- **AccECN**: Change to TCP wire protocol
  - Repeated count of CE packets (**ACE**) - essential
  - and CE bytes (**AccECN Option**) – supplementary

  - Key to congestion control for low queuing delay
    - 0.5 ms (vs. 5-15 ms) over public Internet
Rationale for using TCP flags in SYN (B.1)

- **Backward compatible**
  - server uses latest f/b variant it recognizes
    - no ECN: XXX, RFC3168 ECN: X11, AccECN 111

- **Why use the 3rd LSB?**
  - Had been allocated to ECN nonce sum (NS, now historic→reserved)
  - AccECN combines AE with 2 ECN flags to create 8 codepoints
  - Reserves the nonce-related codepoints for future use

- **If we reserve the 3rd LSB for some future protocol...**
  - The future protocol would not efficiently combine with the 2 ECN flags
  - AccECN would have to use an option on the SYN
    - traversal and space problems
  - AccECN would still have to set the 2 ECN flags for fall-back
    - and deal with all the current middlebox mangling of those 2 flags
    - as well as deal with all the inconsistencies between these 2 flags and the option on the SYN
Rationale for using all 8 codepoints in SYN/ACK (B.2) [since draft-04 (copy of 'bleaching' slide from Nov'17)]

1. **Consumes last 2 combinations of TCP/ECN flags on SYN/ACK**
   - Also protects against ECN-capable proxies blindly forwarding AE flag

2. **Same coding on ACK**
   - Nonce (and possibly 'broken') should become available later
Space for the Future (B.3)

• Future AccECN variants
  • 2 codepoints on SYN/ACK (previous slide)
  • 5 unused codepoints on final ACK of 3WHS
  • 7 unused codepoints on server's 1\textsuperscript{st} data packet
    − note: version negotiation complex on later packets
    − esp. with TFO

• Future non-AccECN uses
  • 5 / 8 codepoints on SYN unused
    001, 010, 100, 101, 110
    − would preclude using any form of ECN at the same time
    − all 8 codepoints on SYN/ACK available in response, except 000 & reflection
  • 3 TCP flags still reserved
    − traversal problems
Generic Receive Offload
(a poor attempt to summarize long ML and offlist discussion)

- during run of CE marks, ACE increments each pkt, preventing merge
- Yuchung would prefer to use DCTCP feedback in TCP header flags, despite indeterminism due to delayed ACKs and pure ACK loss [RFC7560 appendix]
- various suggestions to resolve the dilemma
- otherwise, 3 ways to accommodate:
  1) 2 parallel drafts:
     - current AccECN (with ACE counter)
     - same as AccECN but with DCTCP in TCP header flags, negotiated with TCP option on SYN
  2) Both mechanisms within AccECN draft, selected by initial value of ACE
  3) Change AccECN draft (and code) to use DCTCP in TCP header flags

- committed to work on finding a resolution
Next Steps

- Attempt to resolve GRO issue
- Acknowledge Yuchung's recent contributions to the draft
  - other recent contributors: Praveen & Michael Scharf, are already ack'd
- Address the outstanding issues from Michael Scharf's recent useful additional review comments
  - last para of intro (recommend to complement solely with ECN++)
  - consistency of informative S.2 with recent changes to normative S.3
- WGLC