Explicit Congestion Notification in QUIC

Magnus Westerlund, Ericsson
Based on the Design Team Proposal

Pull Request: https://github.com/quicwg/base-drafts/pull/1372

2018-06-06

Overview

- Overview of Proposal
  - Capability Check
  - ACK_ECN Frame
  - Continuous Verification
  - Congestion Experience Response
  - Connection Migration
- Open Issues
  - ACK Frequency and Recovery Period
  - Packet Duplication
  - ECN Black Hole Mitigation

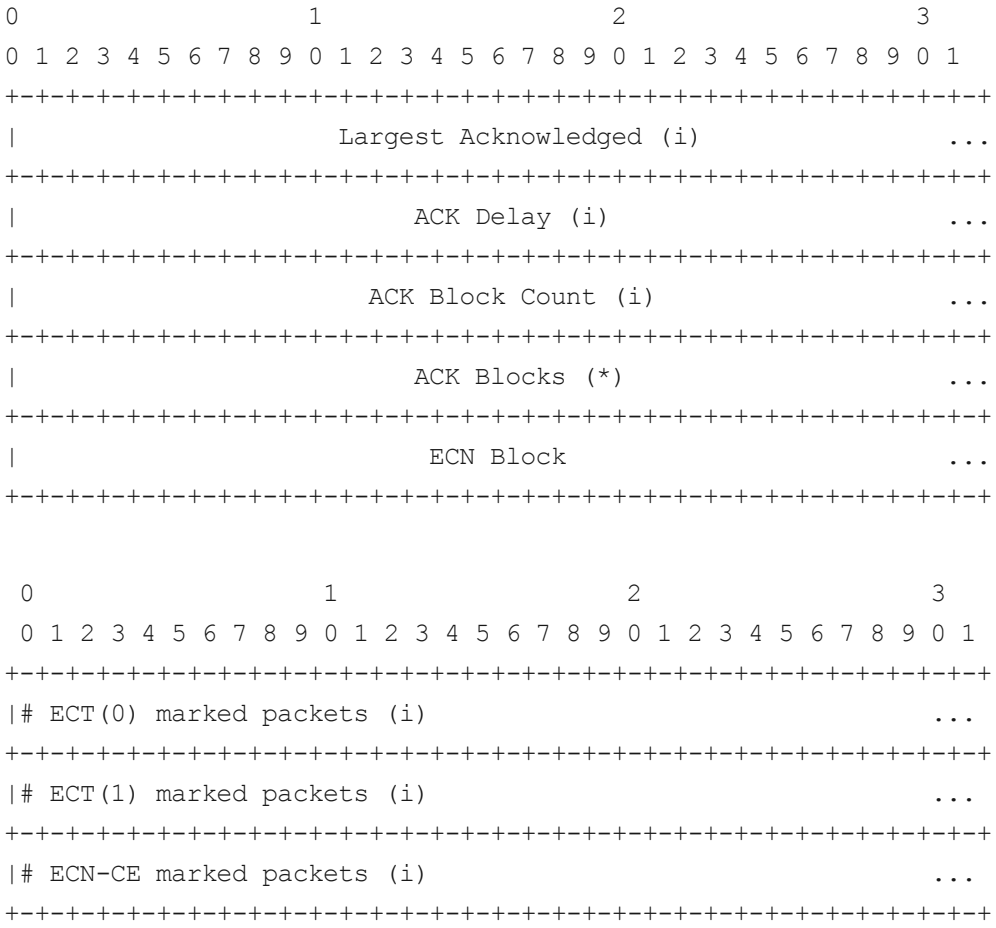
Proposal – ECN Capability Check

- Capability Check to
  - Verify that path don't bleach
  - Verify that sender and receiver are capable of marking and receiving mark
- ECN Capability is handled per send direction
  - Thus, ECN may be used only in one direction
- Started with first packets that may be ACKed
  - Client: Initial Packet
  - Server: Handshake Packet
- Mark all packets sent as ECT
  - ECT(0) is default
  - ECN Experiments may use ECT(1), see RFC 8311

- Terminology:
  - ECT: ECN Capable Transport
  - ECT(0): ECT marking 0, i.e. 10
  - ECT(1): ECT marking 1, i.e. 01
  - ECN-CE: Congestion Experience, i.e. 11
  - Not-ECT: ECN not used, i.e. 00

Proposal – ACK_ECN Frame

- A Variant of the ACK frame
  - Adds Counters for the ECN markings
  - ECT(0), ECT(1), ECN-CE
- Used if endpoint will ACK any packet which was received with ECN markings where
  - ECN field is not Not-ECT
- Counters Cumulative since session start



Proposal – Capability Check Verification

- Sender verifies it send path by
  - Receiving an ACK_ECN frame
  - Calculating total number of marked and ACKed packets
    - Add New ACKed to local counter
  - Calculate Total number of ECN marked packets
    - Sum of the three ACK counters
  - Compare if Total_ECN_marked >= Total_ACKed
    - If True ECN works given that Total_ACKED non-zero
    - Larger than for packet duplication
- Detecting failures
  - Not all packets ECN marked
    - ECN Bleaching
  - Receive ACK rather than ACK_ECN
    - Indicates no packets with markings reached receiver
    - Bleaching
    - Sender incapable of marking
    - Receiver incapable of reading marks
- On failure turn off ECT marking

Proposal – Continuous Verification

- After Successful Initial Capability Check
  - On each ACK_ECN:
    - Calculate Total_ACKed packets and compare with ECN counter
  - Perform comparison
    - If Total_ACKed is bigger than sum of counters then disable ECN
- Implements rather strict behavior in regards to ECN Bleaching
  - Ensures that ECN-CE marks are not missed
  - Avoid favoring connection's packets compared to not-ECT traffic
- Any packet duplications that are recorded in ECN counters will increase insensitivity to bleaching
  - Proposed that any packet duplicates are not included in ECN counters
    - Report first packet only
  - Raised issues that transport draft clarifies duplicate handling in general:
    - https://github.com/quicwg/base-drafts/issues/1405

Proposal – Congestion Experienced

- When receivers receive packet with ECN-CE marking
  - Immediate ack, i.e. not full ACK delay
  - In recovery period normal ACK frequency
- Sender compares ECN-CE counter for each ACK_ECN packet received.
  - If larger than previously stored ECN-CE counter
  - Event at Largest Acknowledged
  - Congestion event
    - Start Recovery Period
    - Reduce CWND same as loss does
      - ECN Experiment to change function

Proposal – Connection Migration

- Connection Migration possibly new path
  - Check if new path is ECN capable
- If ECN was enabled up to connection migration
  - Continue with continuous verification
- If ECN was disabled prior to connection migration
  - Re-run Capability Check
    - Enable ECT on all packets from the first non-probe one
    - Store packet number from where ECT was enabled
    - Store ECT-Counters from latest ACK_ECN and what highest PN was ACKed
- Upon reception of ACK_ECN
  - Check that it ACKs packet after migration was initiated
  - If packet also ACKs packet prior to migration
    - Update stored ECN counters and Highest ACKed PN
  - When only packets after migration are ACKed perform ECN counter comparison
- Reception of ACK for packets after connection migration
  - Disable ECN

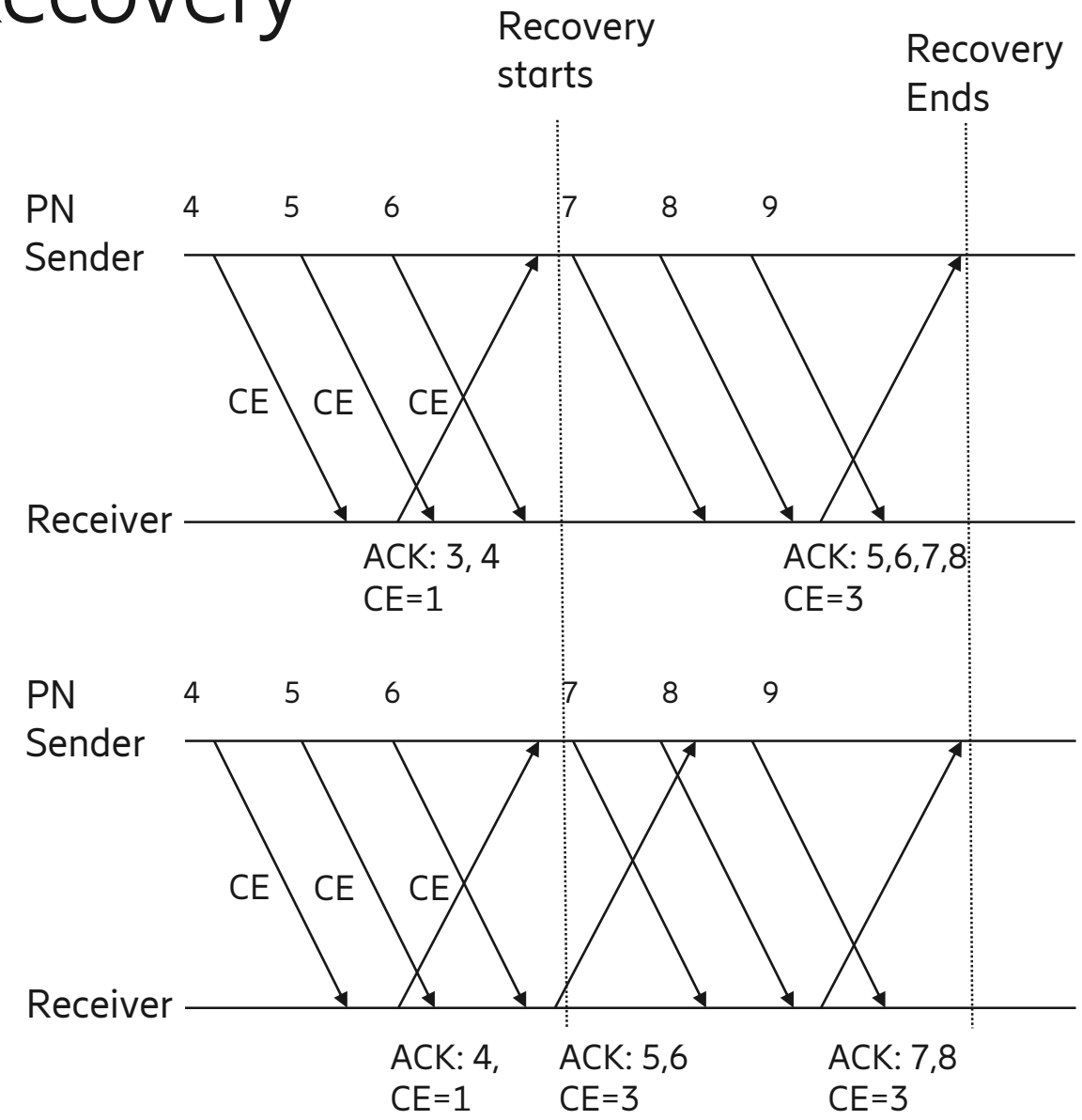
Issues

— ACK Frequency and Recovery Period
— Packet Duplication
— ECN Blackhole Mitigation

Issue: ACK Frequency and Recovery Period

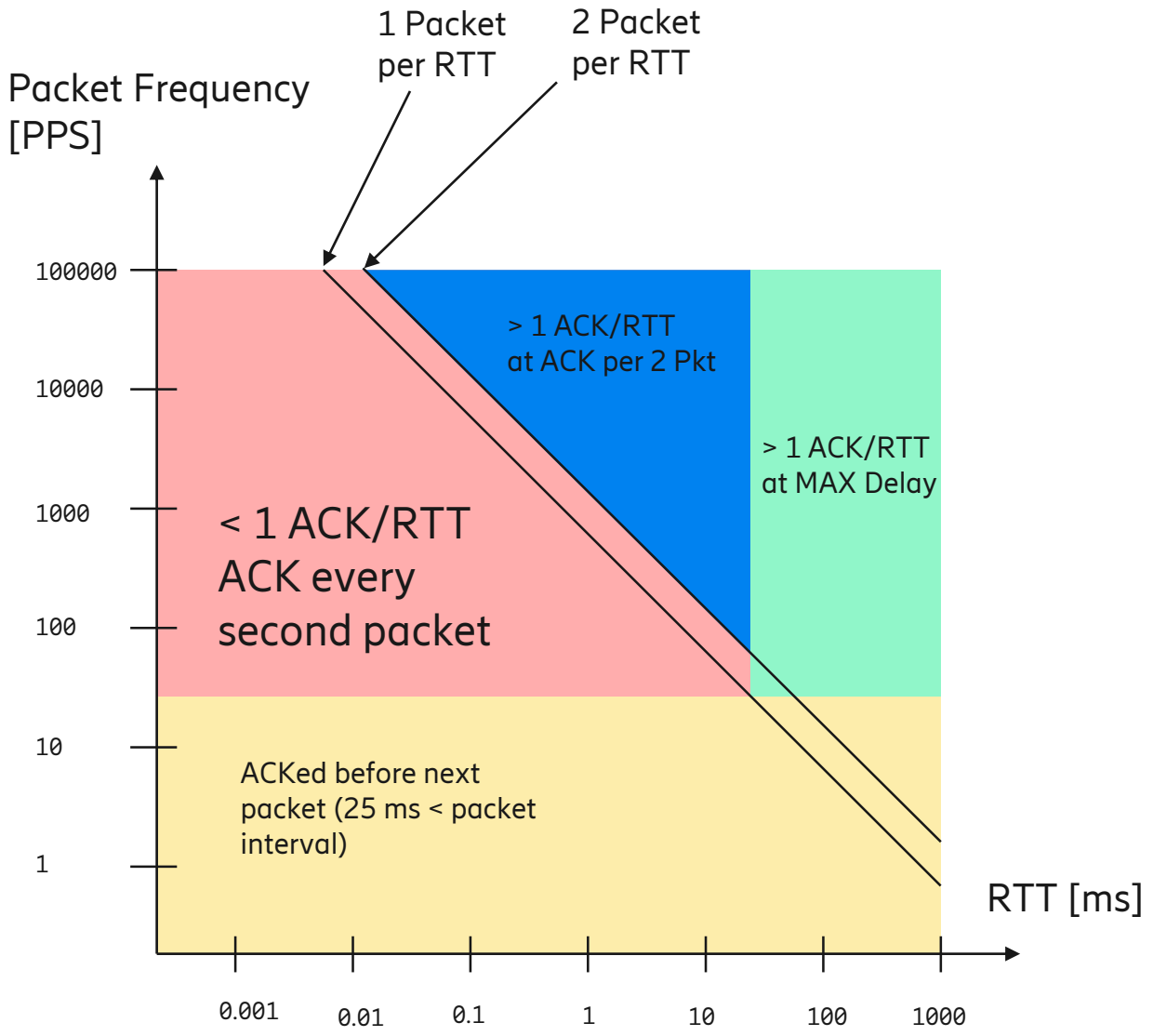
- ECN-CE marks triggers ACK_ECN with increased ECN-CE counter
- Sender goes into recovery period Starting at sending of PN=7
- Recovery ends with ACKing of PN=7
- Uncertainty if CE marks are during recovery?

- If All CE marks in some congestion event are ACKed prior to the ACK ending recovery then no new Congestion Event
- Thus important to have sufficient ACK frequency in regards to RTT



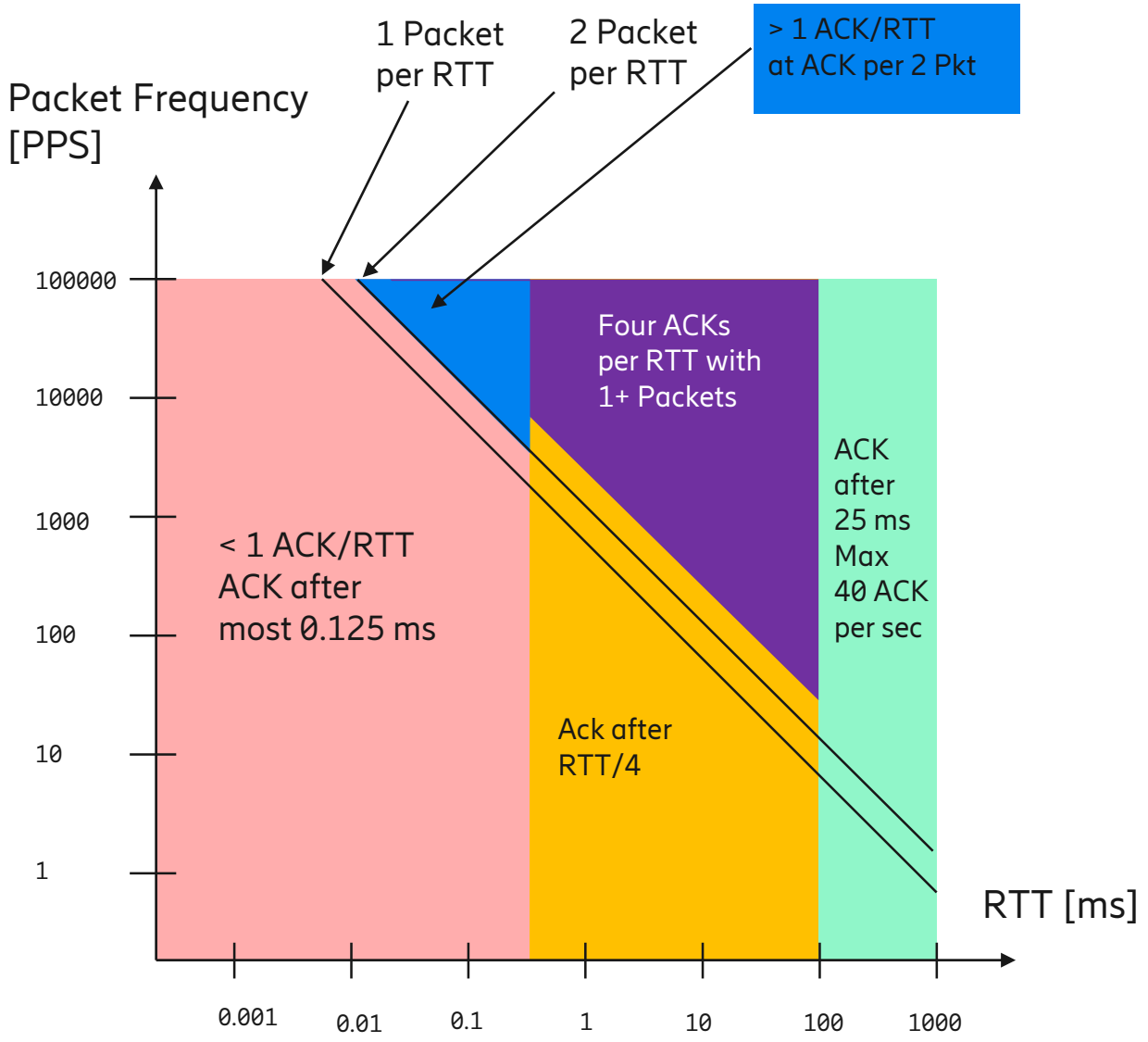
Consideration of QUIC ACK Frequency

- Current ACK rules
  - Max ACK Delay 25 ms
  - May ACK no more often than every 2nd packet unless delayed limit hit
- No Stated consideration of RTT
- Potential issues
  - Low ack rate at LAN RTT (1-5 ms)
    - ECN issues with Recovery for higher Packet Rates
  - For Loopback and Inter Process on same host
    - Allows for optimizations, but either boundary not optimal



Idea: ACK Frequency

- Scale Max_ACK_Delay to min(sRTT/4, 25 ms)
  - Ensures four ACKs per RTT
  - Likely needs clamping at very short RTTs
    - Below 0.5 ms ACK rate becomes high (8000 ACK / Sec)
      - Intra Host Communication range
  - Benefits
    - Ensure ECN does not react twice to a temporary congestion epoch
    - Responsive CWN growth and loss detection at shorter RTT
    - General useful change
  - Downsides
    - Increases number of ACKs in certain areas



Alternative Idea : Receiver side tracking of recovery period

- The receiver knows when it sends an ECN-CE mark, and thus start a recovery period
- Increase ACK frequency from ACK with ECN-CE
  - ACK of ACK likely sent with the packet that starts recovery period
  - Continue with higher frequency until ACK of ACK
- Benefit
  - Would reduce risk of double congestion events
- Downside
  - Extra receiver side processing
  - Uncertainty on when recovery period

Issue: Packet Duplication

- ECN counter are for received markings over all packets so far in connection
- If duplicate packets are counted
  - Increments ECN counters over total of ACKed packets
  - Buffer before detecting ECN bleaching
- Security concern with packet duplicates
  - Enables attacker that can copy valid packets and send replay copies with CE marks
    - Drive down congestion window
    - Without needing to race the original packet
- Proposal
  - Basic packet duplication mechanism
  - Keep basic bit vector or use ACK structure
    - If received packet has PN that matches already received and authenticated packet
      - Drop it!
  - Data structure depth need to be deep enough to at least handle reordering
    - Short RTTs can have larger reordering than min RTT
- https://github.com/quicwg/base-drafts/issues/1405

ECN Blackhole Mitigation

- Sender side only mechanism
  - Thus Optional on what level of complexity versus impact to implement
- Possible Mechanisms
  - Let handshake go into protocol fallback
    - No extra code
  - Retransmit Handshake without ECT
  - Partial marking or probe packets only with ECT marking until one packet verified
    - Then go into continuous verification and mark all

- Refs:
  - https://csperkins.org/publications/2015/10/mcquistin2015ecn-udp.pdf
    - Tested with UDP and TCP towards 2500 NTP servers
    - 98.97% reachability with ECT after Not-ECT
    - 99.54% reachability of Not-ECT after Not-ECT
    - One test location had only ~90% reachability with ECT
  - https://www.bobbriscoe.net/projects/latency/ecn_commag_2018_preprint.pdf
    - Reported no ECN blackholes on 6.5 million different paths

Conclusions

- Have some directions for the issues
- Will address those that are directly related ECN text
- Are there rough consensus to include the functionality in PR?
  - https://github.com/quicwg/base-drafts/pull/1372
  - Capability Check
  - ACK_ECN frame
  - Continuous Verification
  - Classic CE response
- Resolutions to the Issues will be separate Pull Requests