Adding Acknowledgement Congestion Control to TCP

S. Floyd, A. Arcia, D. Ros, and J. Iyengar
draft-floyd-tcpm-ackcc-02.txt
TCPM
December 2007
How would TCP’s ACK Congestion Control work?

- **Negotiation** between sender and receiver:
  - (Ack-Congestion-Control-Permitted option).
- **Start** with an Ack Ratio of 2.
- The sender detects **lost Ack packets**:
  - And tells the receiver the new Ack Ratio.
- **The sender** uses Appropriate Byte Counting and rate-based pacing (in response to Acks acking more than two packets).
Changes from last time:

• Added a section on "Keep-alive Packets". Feedback from Anantha Ramaiah.

• Added a section on "Possible Complication: TCP Implementations that Skip ACK Packets". Motivated by reports at IETF that many high-bandwidth TCPs don't follow the MUST of sending an ACK for every other packet, if they don't have time.

• Added that receivers might have buffer limitations that require that they ack at least every K packets, for some K. Feedback from Sara Landstrom.

• Added to the discussion of "Possible Complication: Two-Way Traffic". Feedback from Sara Landstrom.
More changes from last time:

• Added a section on "Possible Complication: Router or Middlebox-based ACK Mechanisms". Feedback from Sara Landstrom.

• Added that SACK is required with ACK congestion control. Feedback from Sara Landstrom.

• Added a discussion of "Reducing the TCP Acknowledgment Frequency" to the related work section.

• Added an appendix on "Design Considerations", with a subsection on "The TCP ACK Ratio Option, or an AckNow bit in data packets?".

• General editing from feedback from Alfred Hoenes.
Changes in draft-floyd-tcpm-ackcc-03b.txt:

- General editing. Feedback from Alfred Hoenes.

- Added more about keep-alive packets and window update packets. Feedback from Anantha Ramaiah.
Possible Complication: TCP Implementations that Skip ACK Packets

• “One possibility for addressing this problem would be for TCP receivers using ACK congestion control to be required to send an acknowledgement for each $R$ packets, for ACK Ratio $R$.”

• “A second possibility would be to define a TCP option or flag that the TCP receiver could use, when sending an ACK packet, to inform the sender that the TCP receiver `skipped' some ACK packets, so that the sender should not infer ACK loss if some ACK packets seem to be missing.”
Future work:

- **Simulations** and other evaluation of proposed mechanism.

- Ready to be a *working group document*, targeted as Experimental?
Slides from last time:
Possible Complications:

- Delayed acknowledgements.
- Duplicate acknowledgements.
- Two-way traffic.
- Reordering of Ack packets.
- Abrupt changes in the Ack path.
- ...
Congestion on the reverse path:

- Does pure Ack traffic really contribute to congestion?
  - Yes, somewhat, if the queue is in units of packets.
  - Measurement studies of congested links?
- How might ackcc be useful to the connection?
  - ECN-capable ACK packets.
  - Possibly reducing the ACK drop rate even without ECN.
- How might ackcc be harmful to the connection?
  - Costs of a larger Ack Ratio.
Security Considerations:

• Cheating with ECN-capable ACK packets?
  – If the receiver cheats, the sender could detect it.
  – If the sender cheats, the receiver can’t easily detect it.

• Middleboxes probably could detect it.