# Adding Explicit Congestion Notification (ECN) Capability to TCP's SYN/ACK Packets

A. Kuzmanovic, A. Mondal, S. Floyd, and K.K. Ramakrishnan

draft-ietf-tcpm-ecnsyn-02.txt

TCPM

July 2007

# Purpose:

- Specifies a modification to RFC 3168 to allow TCP SYN/ACK packets to be ECN-Capable.
- Based on the SIGCOMM 2005 paper by A. Kuzmanovic.
- Avoids the retransmit timeout when a SYN/ACK packet would have been dropped.
- If the SYN/ACK packet is ECN-marked, the sender of that packet responds by reducing the initial window to one segment, instead of two to four segments.

# More:

- The SYN/ACK packet can be sent as ECN-Capable only in response to an ECN-setup SYN packet.
- The SYN packet still MUST NOT be sent as ECN-Capable.
- The benefit of adding ECN-capability to SYN/ACK packets can be high, particularly for small web transfers.

# The TODO List from March 2006:

- Converge on the response to a marked SYN/ACK packet.
- Look at the costs of adding ECN-Capability in a worst-case scenario. (From feedback from Mark Allman and Janardhan Iyengar.)
- Find out how current TCP implementations respond when receiving a SYN/ACK packet that has been ECN-marked?

# Response to an ECN-Marked SYN/ACK Packet?

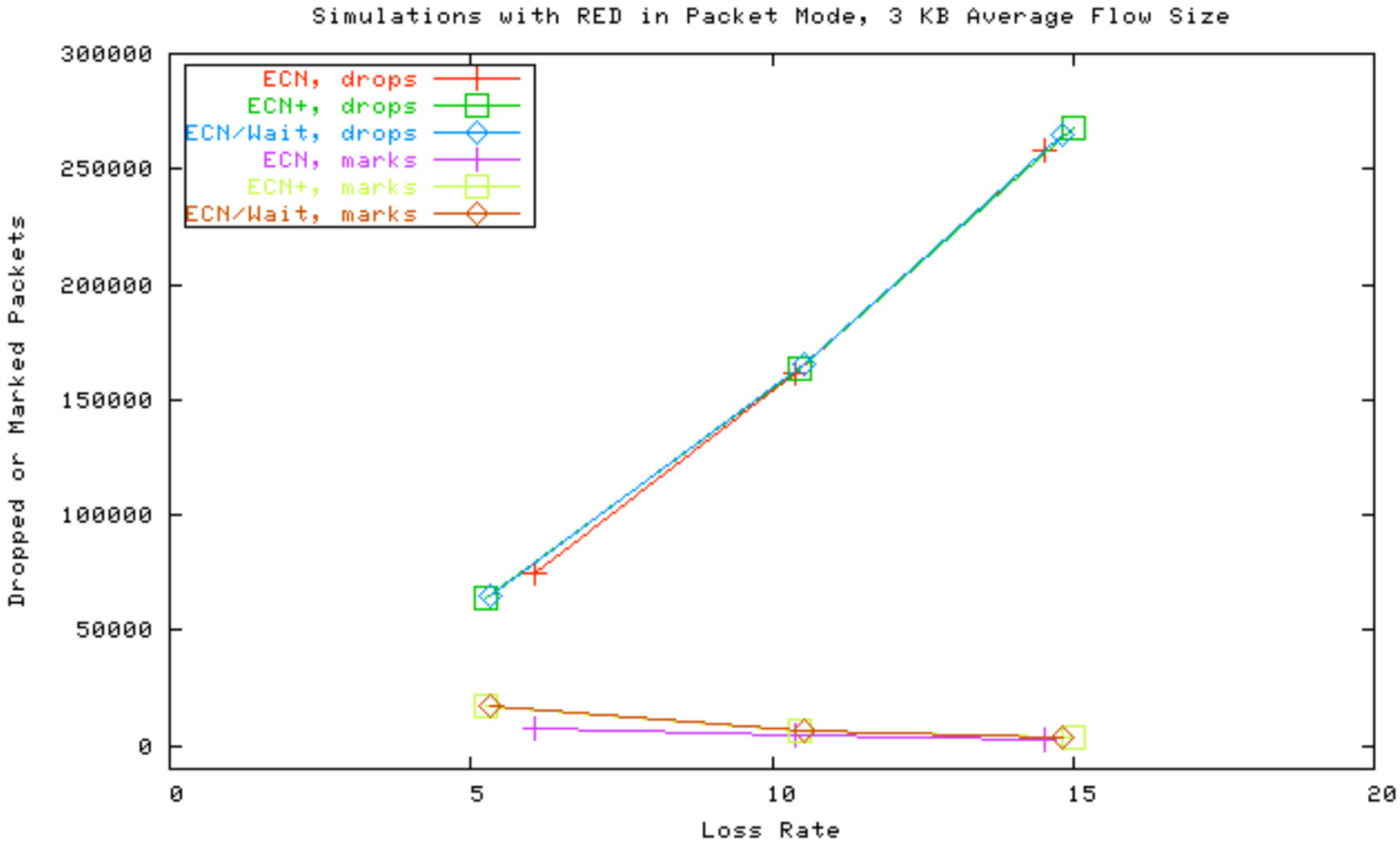
- Set initial cwnd to one packet:
  - Instead of setting cwnd to 2-4 packets.
  - Continue in congestion avoidance instead of slow-start.

OR

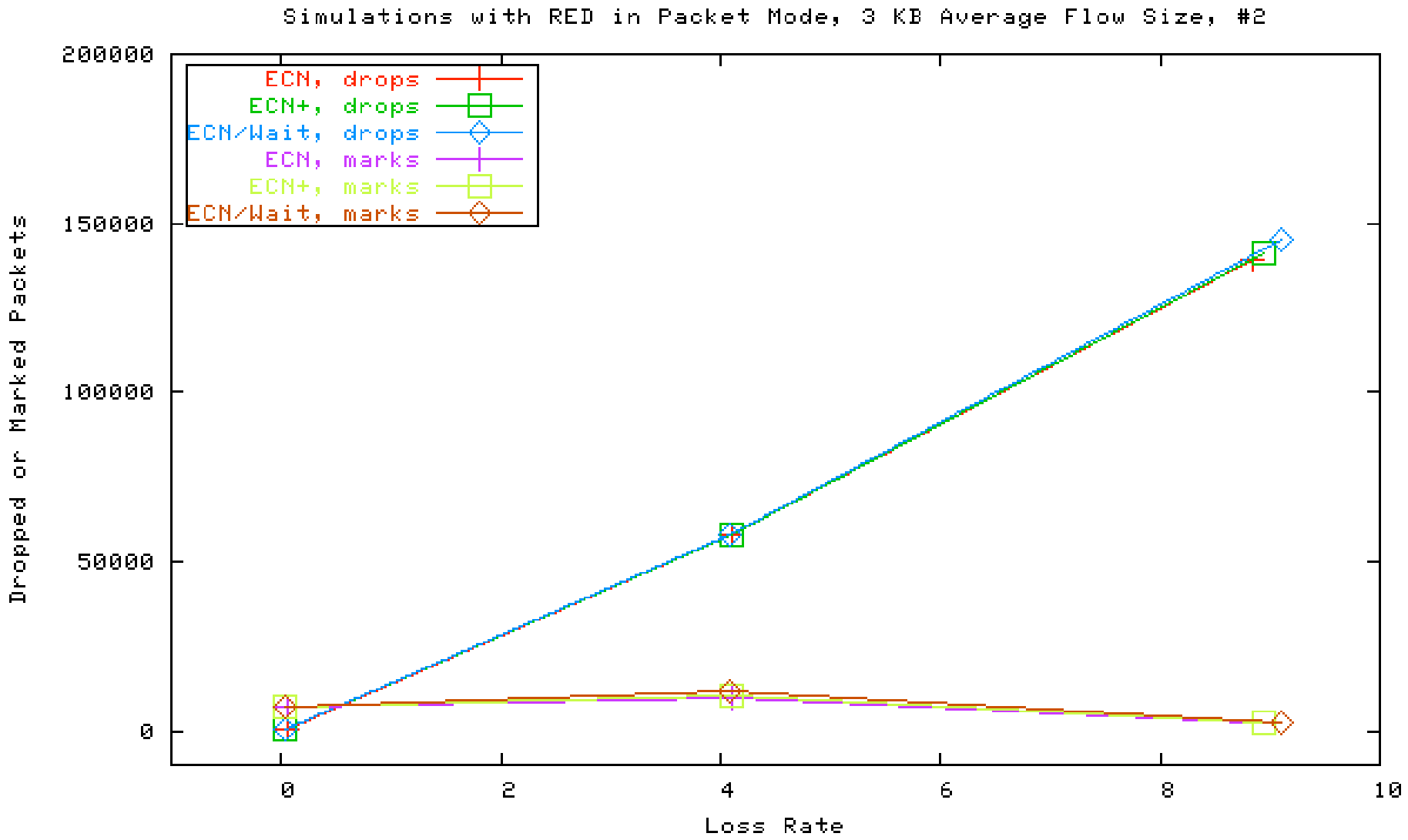
- Wait an RTT before sending a data packet:
  - Proposed by Mark Allman.

- Simulations reported in Appendix A.

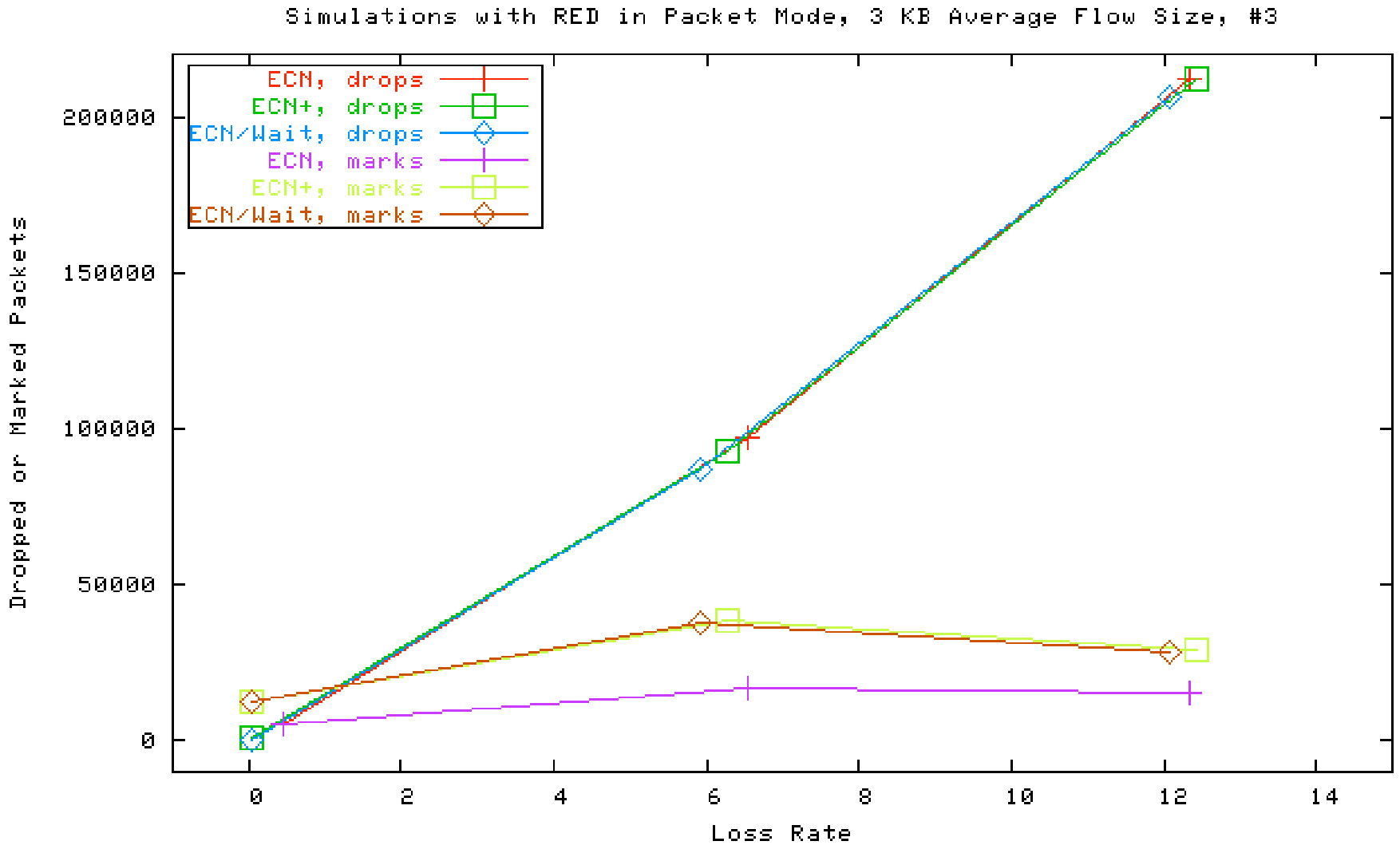
# Results from Simulations:



# Results from Simulations:



# Results from Simulations:



# Simulation Overview:

- Heavy-tailed distribution of file sizes
  - With a range of average file sizes.
- Topology:
  - Target delay 1 ms, 5 ms, 10 ms.
  - 100 Mbps congested link.
  - Minimum RTT of 12 ms.
  - RED in gentle mode.
- Simulations with RED in packet and byte mode.
  - For the simulations with RED in byte mode, SYN packets aren't dropped or marked very often. So it doesn't make much difference if SYN/ACK packets are ECN-Capable.

# Lessons from Simulations:

- Dangers with high congestion?
  - When congestion is high, packets are dropped rather than ECN-marked, with or without ECN+.
- Comparing ECN+ with ECN/Wait:
  - The overall congestion level with ECN+ (without waiting) is similar to that with ECN/Wait (waiting after an ECN/SYN packet is marked).

# Current TCP Implementations:

- Fedora Linux TCP:
  - Shouldn't crash after an ECN-marked SYN/ACK packet.
  - Shouldn't respond to the CE codepoint in a SYN/ACK packet either.
- FreeBSD?
- Microsoft Vista?

# Next steps?

# Extra Viewgraphs:

# Security Concerns:

- "Bad" middleboxes that drop ECN-Capable SYN/ACK packets?
  - We don't know of any.
  - If the first SYN/ACK packet is dropped, the retransmitted SYN/ACK should not be ECN-Capable.
- There is no danger on congestion collapse:
  - Routers are free to drop rather than mark ECN-Capable packets.
  - If the SYN/ACK packet is marked, the sender sends at most one data packet; if that packet is dropped or marked, the sender waits for a retransmit timeout.

# Changes in January (2006) revision:

- Added a discussion to the Conclusions about adding ECN-capability to relevant set-up packets in other protocols. From a suggestion from Wesley Eddy.

- Added a discussion of one-way data transfers, where the host sending the SYN/ACK packet sends no data packets.

- Added a description of SYN exchanges with SYN cookies. From a suggestion from Wesley Eddy.
  - This needs further clarifications.

# The guidelines:

- RFC 3168:

"Upon the receipt by an ECN-Capable transport of a single CE packet, the congestion control algorithms followed at the end-systems MUST be essentially the same as the congestion control response to a *single* dropped packet. For example, for ECN-Capable TCP the source TCP is required to halve its congestion window for any window of data containing either a packet drop or an ECN indication."

- Question:

If TCP's response to a dropped SYN/ACK packet a congestion control response? Or is this a special case, allowing a new response?