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-03.txt
TCPM
December 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.
Changes from draft-ietf-tcpm-ecn-syn-02:

• Added to the discussion in the Security section of whether ECN-Capable TCP SYN packets have problems with firewalls, over and above the known problems of TCP data packets (e.g., as in the Microsoft report). From a question raised at the TCPM meeting at the July 2007 IETF.

• Added a sentence to the discussion of routers or middleboxes that *might* drop TCP SYN packets on the basis of IP header fields. Feedback from Remi Denis-Courmont.

• General editing. Feedback from Alfred Hoenes.
Changes from draft-ietf-tcpm-ecnsyn-03 (not yet submitted):

- **General editing.** This includes using the terms "initiator" and "responder" for the two ends of the TCP connection. Feedback from Alfred Hoenes.
  - URL:
Backwards compatibility issues:

• (1) **Accept problems** with old ECN TCP implementations that don’t respond to ECN-marked SYN/ACK packets?

• (2) Use an **ECN-SYN flag** in TCP header of SYN packet?
  – "I want to use ECN, and I understand ECN-marked SYN/ACK packets”

• (3) Use an **ECN-SYN TCP option**?
  – "I understand ECN-marked SYN/ACK packets.”
Slides from last time:
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:

Simulations with RED in Packet Mode, 3 KB Average Flow Size

Dropped or Marked Packets vs. Loss Rate
Results from Simulations:

Simulations with RED in Packet Mode, 3 KB Average Flow Size, #2

Dropped or Marked Packets vs. Loss Rate
Results from Simulations:

Simulations with RED in Packet Mode, 3 KB Average Flow Size, #3

Dropped or Marked Packets vs. Loss Rate
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?