Adding ECN Capability to TCP’s SYN/ACK Packets

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

draft-ecm-ecn-syn-00.txt

TCPM

March 2006
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.
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 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.
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.
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?
No Congestion:
SYN/ACK Dropped:

![Graph showing SYN ACK Dropped over time and sequence number for Host A and Host B]
SYN/ACK Marked, Response #1:
SYN/ACK Marked, Response #2:
The TODO List:

• 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?
Viewgraphs from last IETF:
Testbed Experiment:

- From Aleksandar’s SIGCOMM 2005 paper on “The Power of Explicit Congestion Notification”.
Testbed Experiments

Client

10 Mbps requests

15 Mbps responses

router 100 Mbps Server pool

Server

3 servers

ECN

no ECN

ECN

ECN+

Client → Server pool

Server pool → Server

Server → Client
ECN and Flash Crowds

<table>
<thead>
<tr>
<th></th>
<th>Average Response Time</th>
<th>Throughput (% of capacity)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RED, no ECN</td>
<td>26 sec</td>
<td>44%</td>
</tr>
<tr>
<td>RED, ECN</td>
<td>4.5 sec</td>
<td>56%</td>
</tr>
<tr>
<td>RED, ECN+</td>
<td>0.5 sec</td>
<td>99%</td>
</tr>
</tbody>
</table>

Reasonable performance despite huge congestion.
Details of testbed experiment:

- 15 Mbps arrival rate, 10 Mbps service rate.
- Very short transfers.