Quick-Start for TCP and IP

Draft-amit-quick-start-04.txt
A. Jain, S. Floyd, M. Allman, and P. Sarolahti
TSVWG, March 2005

Presentation last IETF:
www.icir.org/floyd/talks
QuickStart with TCP, for setting the initial window:

- In an IP option in the SYN packet, the sender's desired sending rate:
  - Routers on the path decrement a TTL counter,
  - and decrease the allowed sending rate, if necessary.

- The receiver sends feedback to the sender in the SYN/ACK packet:
  - The sender knows if all routers on the path participated.
  - The sender has an RTT measurement.
  - The sender can set the initial congestion window.
  - The TCP sender continues using normal congestion control.

- From an initial proposal by Amit Jain
Changes from draft-amit-quick-start-03.txt:

- Added a citation to the paper on "Evaluating Quick-Start for TCP", and added pointers to the work in that paper.
  - Discussions of router algorithms.
  - Discussions of sizing Quick-Start requests.
- Added section on "Misbehaving Middleboxes".
- Added section on "Attacks on Quick-Start".
“Evaluating Quick-Start for TCP”

• **Router algorithms:**
  – Minimal algorithms at routers.
  – Also “**Extreme Quick-Start**” -
    • Maintains per-flow state for Quick-Start flows

• **Sizing the Quick-Start request.**
  – Problems with overly-large Quick-Start requests.
  – Heuristics end-nodes could use in sizing requests.

• URL “http://www.icir.org/floyd/quickstart.html”
Attacks on Quick-Start:

• **Attacks to increase router’s processing load:**
  – Easy to protect against - routers ignore Quick-Start when overloaded.

• **Attacks with bogus Quick-Start requests:**
  – Harder to protect against.
  – Extreme Quick-Start in routers can help..
Misbehaving Middleboxes:

- Traffic normalizers that rewrite IP TTLs along the path?
  - Interferes with Quick-Start mechanism for validating a Quick-Start request.
Feedback?

- Are we ready for Working Group Last Call?
- Experimental?
Extra viewgraphs:
Heuristics for Sizing Quick-Start Requests:

• The sender doesn’t necessarily know the amount of data to be transmitted.
• The sender knows more after an idle period.
• **End-hosts might know:**
  – The capacity of last-mile hop.
  – The size of the local socket buffer.
  – The receiver’s advertised window.
  – Information from the application.
  – Past history of Quick-Start requests.