Eliminating Aborted Data Delivery Over Cellular Links

Andrei Gurtov (gurtov@icsi.berkeley.edu)
University of Helsinki/ICSI

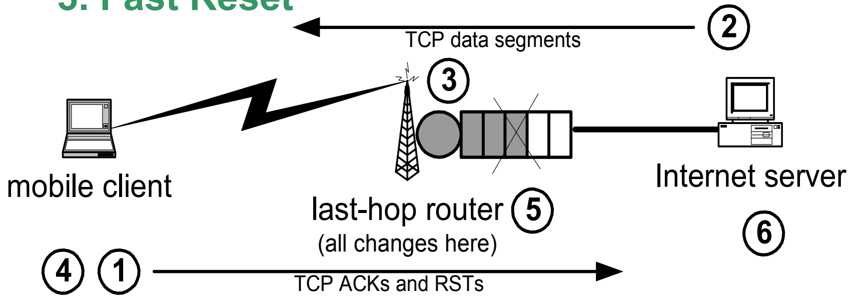
1. Motivation

- Cellular wireless links are still slow and expensive
- Packets from aborted transport connections are delivered unnecessarily over the wireless link
- Analysis of backbone Internet traces show that 15-30% of all TCP connections are aborted
- The average length of completed and aborted connections was 12 packets; resets are not merely connection refusals
- Due to higher response times over cellular links, users get impatient easily; up to 50 % of received data is aborted
- WWW is the dominant generator of aborted data
  - Clicking "Stop", "Reload", "Back", or another web link

2. The Idea

- TCP receiver generates RST packets after receiving and discarding segments from aborted connections
- The last-hop router intercepts RST packets and discards buffered segments from aborted connections
- Works for any TCP application such as HTTP, FTP, peer-to-peer
- A similar implementation in the mobile client for uplink TCP transfers
- Can be adopted for other connection-oriented transport protocols such as SCTP and DCCP

3. Fast Reset



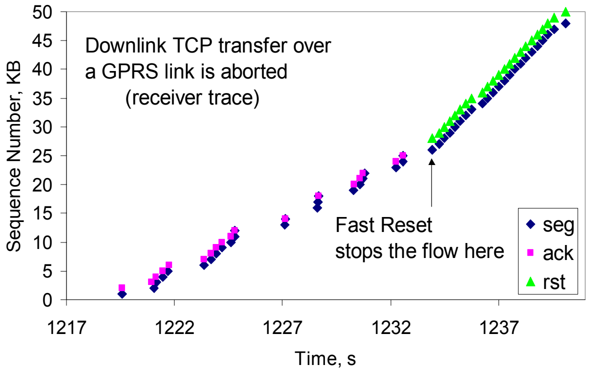
1. Application at the mobile client opens a TCP connection and requests a web object
2. Server receives the request and starts transmitting data to client
3. Data packets are buffered in the access router and transmitted to the client
4. The user decides to abort the download, for example by pressing a 'Reload' button. TCP receiver sends RST packets to the server
5. The last-hop router notices a RST and discards buffered packets in the downlink direction that belong to the aborted connection. It then forwards RST toward the server.
6. Server receives RST and stops transmitting data on the aborted connection.

Effect of Link Buffer Size

- Overhead of aborted data depends on the link buffer size
- Active Queue Management can keep the average queue size low
- Cellular links require a buffer about 2*bandwidth*delay for efficient ARQ

Another Layering Violation?

- Yes, the router has to examine transport-layer headers
  - But Fast Reset is as useful an optimization as header compression
- No layering violation if the layer-3 is connection-oriented as in ISO CONP



Lost RST packets

- What if a RST packet gets lost?
  - The sender would retransmit a data segment after a retransmit timeout
- The Fast Reset algorithm does not keep state of aborted connections
  - A newly arriving data packet is forwarded to the receiver and generates another RST
  - No harm to new TCP connections if the client reuses ports from aborted connections
- In summary: Fast Reset is robust to packet losses and sender's misbehavior

4. Evaluation

- Measurements in Linux using Netscape over GPRS (30 kbps)
- An abort generates 5-30 RST packets distributed over 1-4 TCP connections
  - 1-10 sec is wasted before a new web page starts loading
- With Fast Reset, 1-2 RST packets are sufficient
  - Response time can be reduced by 20 %
  - Battery of the mobile terminal is preserved
  - Money savings (billing is according to data volume)

5. Future Work

- TCP resets are not adequately represented in Internet research
- Most importantly, HTTP traffic generators should be extended to include aborts
  - One approach is to select an appropriate distribution of thinking time, that can also have negative values
- More data at http://www.cs.helsinki.fi/~gurtov