CS551
Handoff Performance in Cellular Networks
[Balakrishnan95b]

Bill Cheng

http://merlot.usc.edu/cs551-f12
Key Ideas

- Deals with TCP in mobile environments
  - packet loss (corruption)
  - handoff (changing from one base station to another)

- Snoop
  - base stations cache TCP segments and quickly retransmit

- Handoff
  - cache TCP segments at nearby base-stations to allow rapid handoff
Problem: TCP Loss Handling in Wireless

- TCP assumes loss implies congestion
  - TCP’s reaction: reduce sending rate

- Wireless adds losses due to corruption, collision, handoff
  - desired reaction: retransmit lost packets quickly

Approach:
- let base-station help out
- alternative is to do link-level reliability
Alternatives

Split-connection TCP:
- from BS, use one TCP connection to FH and another to MH
- but requires changes to FH, BS, MH
- what does an ACK mean now?

Make TCP distinguish congestion vs. other kinds of loss
- good idea: done with ECN
- but done after this work and not widely deployed even today
- requires changes to FH and MH

Link-layer retransmission
- good idea, but must be careful to avoid interactions between link-layer and TCP (works if on different timescale)
Constraints

- Incremental deployment
  - Solution should not require modifications to fixed hosts
  - If possible, avoid modifying mobile hosts

- Preserve TCP end-to-end semantics
  - ACK of a packet means it’s at the receiver, not the base station
Snoop Overview

Base Station (BS) *snoops* passing traffic (data/acks); quickly retx’s data

**FH-to-MH:**
- Fixed Host (FH) sends data to MH
  - no change to FH code
- MH receives data, sends ACKs as usual

**MH-to-FH:**
- BS adds SACK support (even if FH doesn’t support it)
- Data flows from MH to FH
- ACKs flow from FH to MH

Copyright © William C. Cheng
FH-to-MH Snoop Data Processing

1. Packet arrives from FH
   - New pkt?
     - Yes
       - In-sequence?
         - Yes
           - 1. Cache packet
              2. Forward to mobile
             Common case
         - No
           - 1. Mark as congestion loss
              2. Forward packet
             Congestion loss
     - No
       - 1. Forward packet
          2. Reset local retransmission counter
Sender Retransmission

2. Out of sequence, cached
   - In-sequence?
     - Yes
       - New pkt?
         - Yes
           - Add to cache and pass on
         - No
           - Should not be common
   - No
     - Greater than last acked:
       - Pass on
     - Else: generate ACK to fixed host (may be caused by a lost ACK)

3. Out of sequence, not cached
   - Lost or delayed out-of-order
   - Pass on, and keep information
Snoop ACK Processing

1. Ack arrives from MH
   - New ack?
     - Yes
       - 1. Free buffers
         - 2. Update RTT estimate
         - 3. Propagate ACK to sender
     - No
6. Clean up cache
   - Pass on to FH

2. Duplicate ACK
   - Dup ack?
     - Yes
       - Retransmit lost packet with high priority
       - Spurious ack
     - No
       - Discard
6. If data not in cache, or sender retransmit, pass on to FH (not in flowchart)
   - No
     - Discard
   - Yes
     - First one?
       - Yes
         - Retransmit lost packet with high priority
         - Next packet lost
       - No
         - Discard
6. If in cache, respond immediately
   - suppress other dupacks
Handoff Support

General approach:
- extend mobile IP to *multicast* packets to several FA’s (base stations, BSes)
- MH informs BS when it’s changing
- BSes are pre-loaded w/data, can run snoop and quickly repair losses
Other Issues

- What about mobile-to-fixed communication?
  - Modify snoop module to generate SACKs

- TCP over ad-hoc networks?
  - Open area of research
Discussion

Impact
- deployable solution for wireless performance enhancement

Does this violate the end-to-end argument?

Other examples?
- fast-retransmit in TCP
- layer-4 caching? (i.e., caching HTTP without the end points knowing it)

Nice aspects of Snoop
- minimal changes to improve performance
- soft-state design
- preserves TCP semantics
- implementation