# Improving reliable transport and handoff performance in cellular wireless networks

Hari Balakrishnan, Srinivasan Seshan and Randy Katz

Presented by

Eric Hall

# Goals

- Improve end-to-end performance over wireless links w/o changing TCP implementations at hosts *in the fixed network*.
- And avoid recompiling or relinking existing applications.

# Problems with previous works

- I-TCP: breaks TCP semantics, requires relinking, too much overhead, and handoff latency.
- Fast Retransmit:  Good with handoff, but doesn’t handle error characteristics of wikjreless link.

# How does they achieve these goals?

- Modify base stations to monitor traffic and handle wireless transmission, handoff related losses.
- Use multicasting to get packets to more than one base station.

# But first…

- Questions John?

# The Snoop Module

- Processes packets at base station.
- Snoop_data(): processes data packets from fixed host (FH)
- Snoop_ack(): monitors ACK's sent by mobile host (MH)

# Snoop_data()

- Caches packets sent by FH.
- Keeps track of last sequence number seen for connection.
- Three cases for how Snoop_data() handles incoming packets.

# Snoop_data() cont.

Case 1: New packet in normal TCP sequence.

Action: Add the packet to the cache and forward it.

Case 2: Out of sequence packet arrives which is already cached.

Action: If sequence number > last ACK, forward. Else, if sequence number < than last ACK, generate ACK and don't forward packet.

# Snoop_data() cont.

Case 3: Uncached out-of-sequence packet.

Action: Forward to MH, cache and record as having been retransmitted by sender. This is to be used by Snoop_ack().

# Snoop_ack()

- Processes and monitors ACK's sent by MH
- Three cases for how Snoop_ack() handles ACK's

# Snoop_ack() cont.

Case 1: New ACK.

Action: Remove acknowledged packets from snoop cache. If no retransmissions in current window, update RTT estimate. Forward ACK to FH.

Case 2: Spurious ACK.

Action: Discard.

# Snoop_ack() cont.

Case 3: Duplicate ACK, i.e. same as ACK just seen. (DUPACK)

Case 3.1: DUPACK not in cache, or already retransmitted by sender (see 3 slides ago).

Action: Forward DUPACK to FH

Case 3.2: Unexpected DUPACK.

Action: Retransmit lost packet at high priority.

# Snoop_ack() cont.

Case 3.3: Expected DUPACK. In previous case, snoop estimated how many more DUPACKs would be sent.

Action: Discard DUPACK.

# Data transfer from a mobile host

- Need to catch wireless transmission errors from the MH as well.
- Base station monitors data packets transmitted from host. NACKs those that are lost.
- NACKs implemented using SACKs.
- Requires MH be modified to enable SACK processing.

# Routing Protocol(1)

- Utilizes a home agent (HA), and multicast routing.
- Each MH has a long-term IP address for its HA.
- Each MH also receives a temporary multicast address.
- Members of the multicast group are base stations within vicinity of MH.

# Routing Protocol (2)

- MH gathers statistics on signals of beacons for nearby BSs.
- Uses these statistics to determine which BS to join next, where it might go in future.
- Based on this MH configures routing between HA and BSs.
- Primary BS and nearby BSs are all part of multicast group. Primary forwards the packets, all others cache them.

# Handoff

- The handoff is initiated by the MH.
- Done when the host discovers a BS with stronger signal, which it has not been to recently.
- MH sends control messages to deactivate forwarding in old BS and activate it in new BS
- Typically occurs between to BSs belonging to MH's multicast group.
- Implicit assumptions: The cells are overlapping, equal available bandwidth at each cell.

# What to do at handoff?

- Problem: New BS's can't snoop ACKs before handoff
- Solution: Transfer state between BSs?

# What to do at handoff?(2)

- Too much overhead involved in transferring state.
- Instead, new BS uses first received ACK to synch up its contents with those of the previous base station.
- Could it possibly synch up sooner by being promiscuous?

# Implementation of Snoop

- Snoop cache is a circular buffer of packets.
- Size in general is large enough to handle max window size. But preference given to older packets.
- Use reference counting mechanism to avoid copying buffers.
- Maintain RTT (from BS to MH) timer to retransmit packets fast if no acknowledgements are received in 2*RTT.
- Persist timer triggers retransmission if unACKed packets exist in buffer after 200ms of inactivity.

# Performance

- Tested using TCP Reno (fast retransmit). Throughput with no errors was 1.6 Mbps.
- Used Poisson distributed error model.
- Implementation performed significantly better than regular TCP for bit error rates greater than 1 error per 2 Mb of data. At high error rate, up to 20 times better.
- Maximum performance limited to 1.45 Mbps, due to presence of beacons.

# What does this paper give us?

- Perhaps a best case scenario of TCP performance over wireless links and during handoffs.
- Problems:

-Relies on multicasting.

-Same triangulation issue as Mobile IP.

-Simple testing.