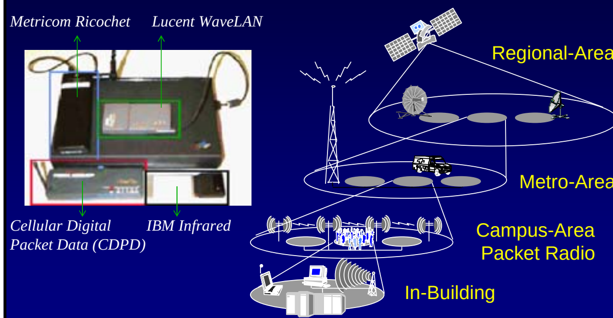
EEC173B/ECS152C, Spring 09

Reliable Data Transport over Wireless Networks

- Problems with TCP
- Snoop Protocol

* Acknowledgment: Slides from Prof. Hari Balakrishnan & Prof. Badri Nath

Wireless Heterogeneity



Wireless Performance

Technology	Rated Bandwidth	Typical TCP Throughput
IBM Infrared	1 Mbps	100-800 Kbps
Lucent WaveLAN	2 Mbps	50 Kbps-1.5 Mbps
Metricom Ricochet	100 Kbps	10-35 Kbps
Hybrid wireless cable	10 Mbps	0.5-3.0 Mbps

Goal: To bridge the gap between perceived and rated performance

Data Transport Over Wireless

- Packet loss in wireless networks may be due to
  - Bit errors
  - Handoffs
  - Congestion (rarely)
  - Reordering (rarely, except in mobile ad hoc networks)

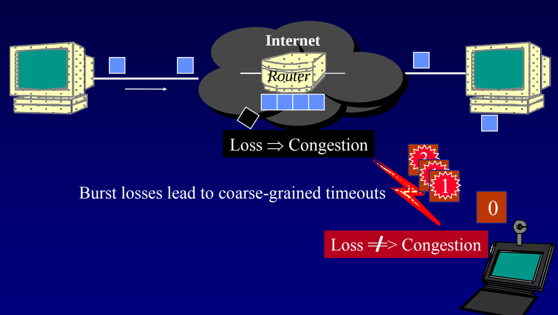
Poor Interaction with TCP

- TCP assumes loss is due to congestion or reordering
- Wireless loss is not due to congestion
  - TCP cannot distinguish between link loss and congestion loss => result in lower throughput
- Cumulative ACK not good with bursty losses
  - Missing data detected one segment at a time
  - Duplicate ACKs take a while to cause retransmission
  - TCP Reno may suffer coarse time-out -> slow start!
  - TCP New Reno still only retransmit one packet per RTT
- Non-congestion loss indicated by DUP ACKs
  - Fast retransmit & recovery (congestion window is halved)
- Non-congestion loss indicated by timeout
  - Enter slow start (Start from CongWin = 1)

Other Problems in Wireless Networks

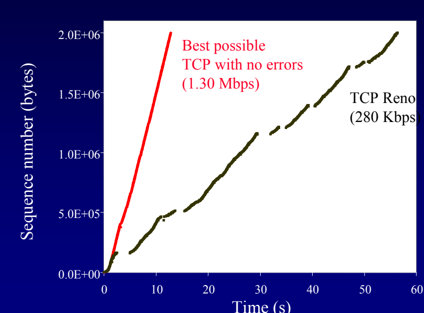
- (#1) Burst errors due to poor signal strength or mobility (handoff)
  - More than one packet lost in TCP window
- (#2) Asymmetric effects
  - Bandwidth asymmetry & latency variability
- (#3) Low channel bandwidth
- (#4) Delay is often very high
  - RTT quite long (tunneling, satellite)
  - True in telephone networks providing data services that deploy fixed gateways (non-optimal routes)

Challenge #1: Wireless Bit-Errors



Result: Low throughput

Performance Degradation



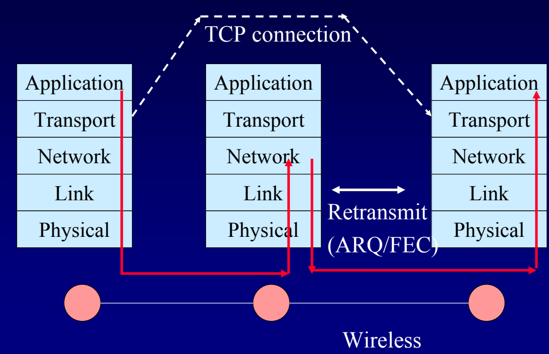
2 MB wide-area TCP transfer over 2 Mbps Lucent WaveLAN

Approaches

Question: how to reconcile between the two in an end-to-end transport mechanism?

- Link layer enhancement (FEC, retransmission)
  - **[LR99]** R. Ludwig and B. Rathony, "Link Layer Enhancements for TCP/IP over GSM," *IEEE Proc. Infocom*, pp. 415-422, 1999.
- Transport Layer
  - **[BB95]** A. Bakre and B. R. Badrinath, "I-TCP: Indirect TCP for mobile hosts," *Proc. 15th International Conference on Distributed Computing Systems*, Vancouver, Canada, June 1995, pp. 136-143.
- TCP-aware Link-layer aware
  - **[BSK95]** Snoop protocol
- Explicit Loss Notification Schemes

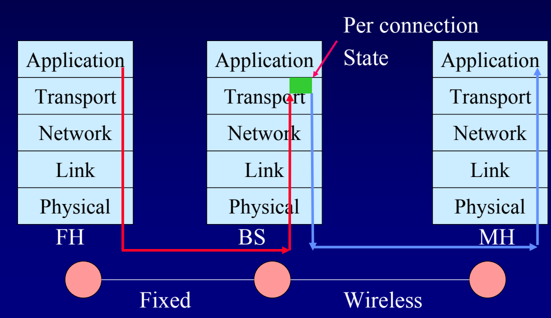
Link Level Retransmission



Link Level Retransmission: Issues

- How many times to retransmit at the link layer before giving up?
- How much time is required for a link layer retransmission?
  - Only beneficial if TCP timeout large enough to tolerate additional delays due to link level retransmission
- What triggers link level retransmission?
- Adverse interaction with transport layer
  - Timer interaction
  - Interaction with fast retransmit
  - Large variation in RTT

Transport-level Solution



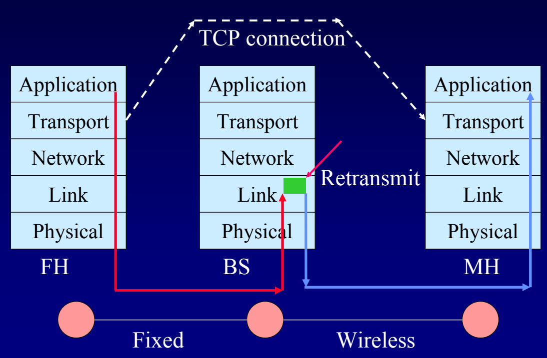
I-TCP

- Split end-to-end connection into two independent flows
  - One connection for the wired part, and another for the wireless part
  - Wireless part of the TCP can be optimized for wireless
    - Different flow/error control
    - Local recovery of errors: faster recovery due to shorter RTT on wireless link
  - On wireless, loss -> try harder
  - On fixed, loss -> backoff

I-TCP Disadvantages

- End-to-end semantics violated
  - ACK may be delivered to sender before data delivered to receiver
- Base station (BS) retains hard state; its failure can result in loss of data (unreliability)
- BS retains per-connection state -> not scalable
  - Buffered packets at BS must be transferred to new BS
  - Buffer space needed
- Hand-off latency increases due to state transfer
  - Extra copying of data at BS

Snoop [BSK95]: TCP-aware, Link-aware



Snoop Protocol

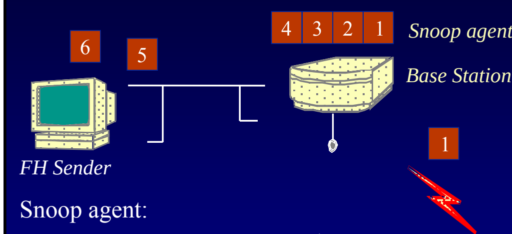
- Uses the same idea of local recovery as I-TCP
- Shield TCP sender from wireless vagaries
  - Eliminate adverse interactions between protocol layers
  - Congestion control only when congestion occurs
- Preserve current TCP/IP service model
  - Maintain end-to-end semantics

Fixed to mobile: transport-aware link protocol
Mobile to fixed: link-aware transport protocol

Snoop Features

- Snoop monitors every packet that passes through
  - Buffers packets from FH to MH as yet unacknowledged
  - Packets flushed when an ACK is received
  - When DUP ACK is received, retransmit from buffer
- Hide wireless loss from sender
  - Suppress DUP ACKs => prevent fast retransmit
  - Sender can still timeout
- Snoop state is soft state at base station, instead of hard state
  - Handoff -> new snoop state is built at new BS
  - Loss of soft state affects performance, but not correctness

Snoop Protocol: FH to MH

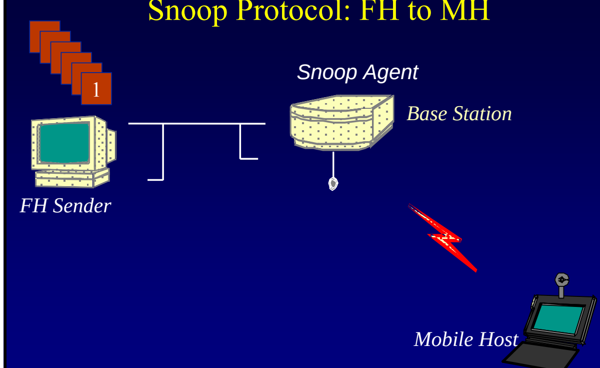


Snoop agent:

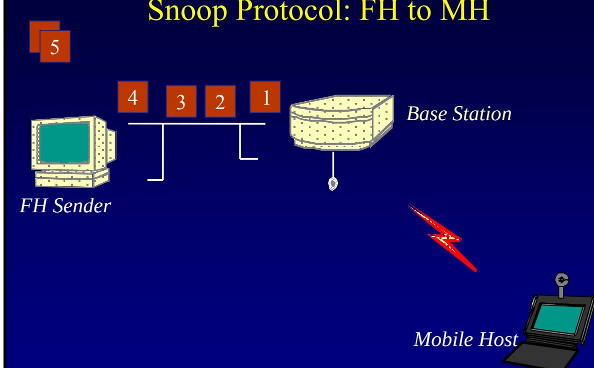
- Snoops on TCP segments and ACKs
- Detects losses by duplicate ACKs and timers
- Suppresses duplicate ACKs from FH sender

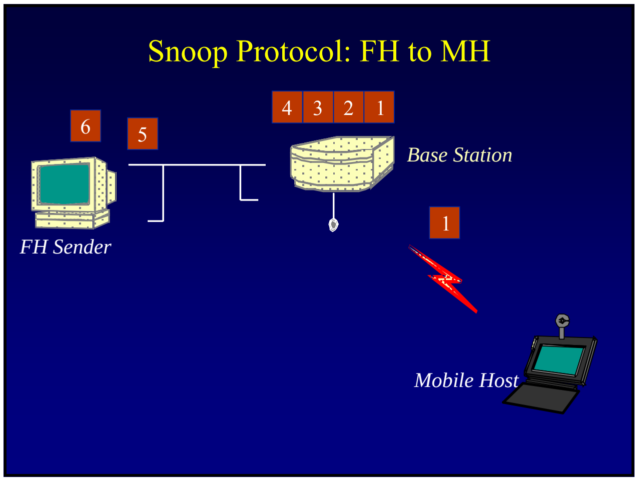
Cross-layer protocol design: Snoop agent state is *soft*

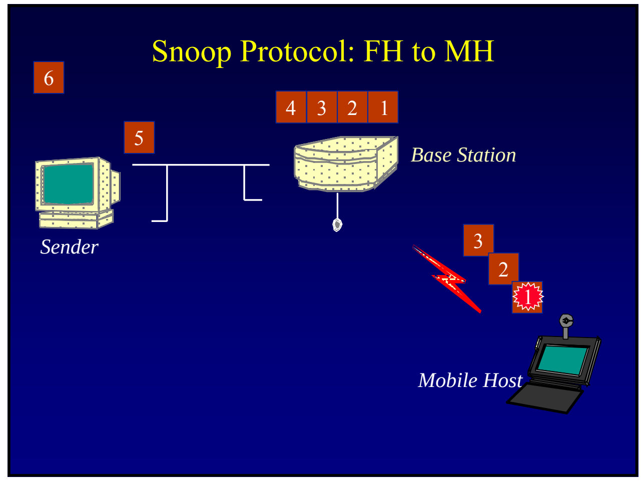
Snoop Protocol: FH to MH

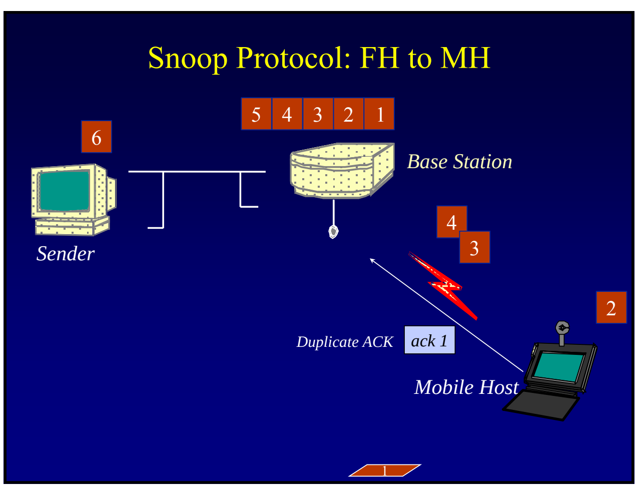


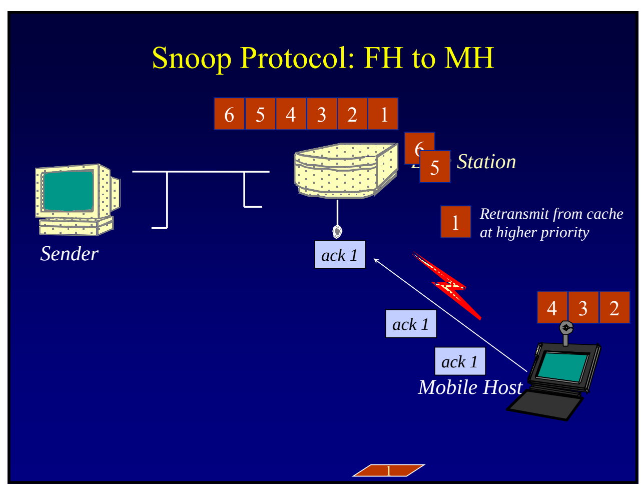
Snoop Protocol: FH to MH

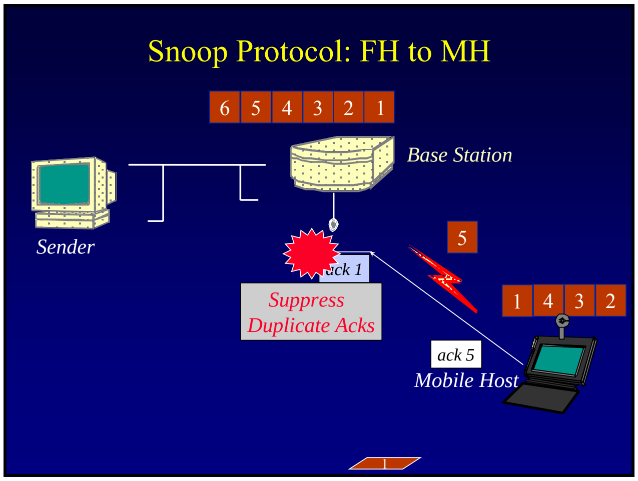


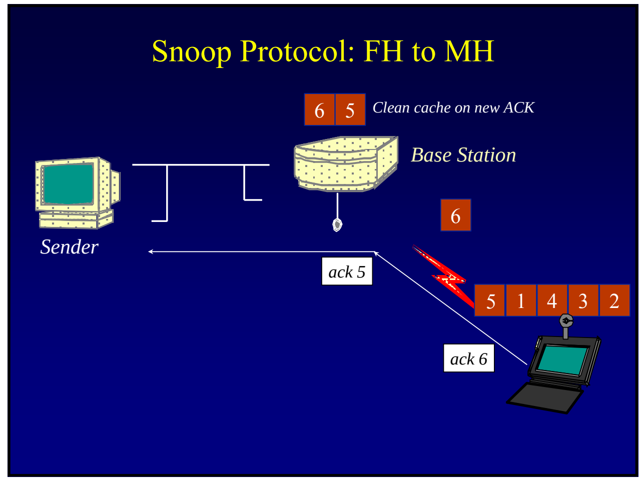

Snoop Protocol: FH to MH
6
5
4 3 2 1
Base Station
FH Sender
1
Mobile Host

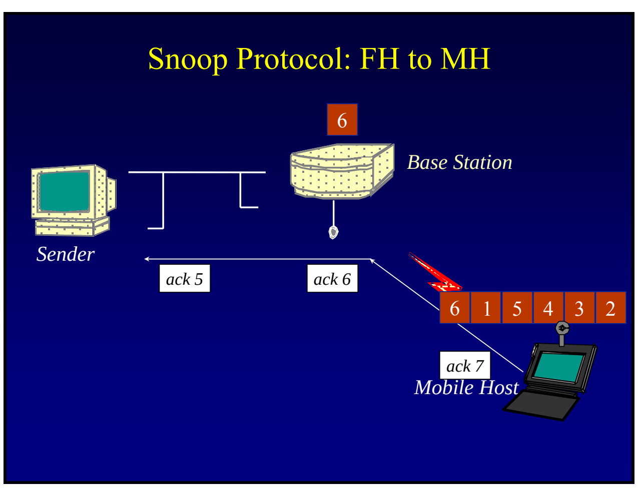

Snoop Protocol: FH to MH
6
5
4 3 2 1
Base Station
Sender
3
2
1
Mobile Host

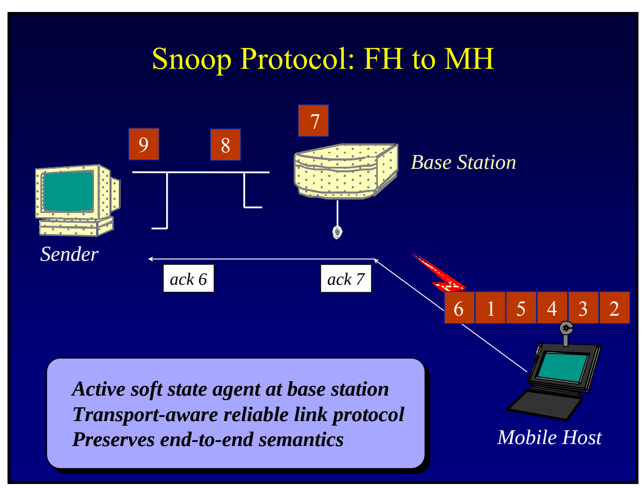

Snoop Protocol: FH to MH
5 4 3 2 1
6
Base Station
Sender
4
3
2
Duplicate ACK
ack 1
Mobile Host


Snoop Protocol: FH to MH
6 5 4 3 2 1
5
Station
Sender
1
Retransmit from cache
at higher priority
ack 1
4 3 2
ack 1
ack 1
Mobile Host

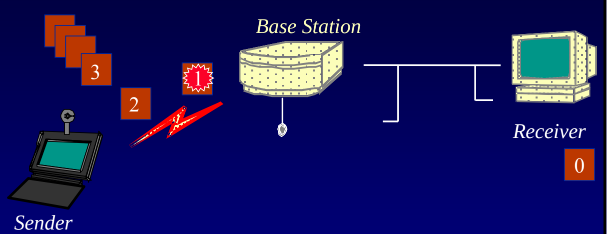

Snoop Protocol: FH to MH
6 5 4 3 2 1
Base Station
Sender
5
Suppress
Duplicate Acks
1 4 3 2
ack 5
Mobile Host


Snoop Protocol: FH to MH
6 5
Clean cache on new ACK
Base Station
Sender
6
ack 5
5 1 4 3 2
ack 6


Snoop Protocol: FH to MH
6
Base Station
Sender
ack 5
ack 6
6 1 5 4 3 2
ack 7
Mobile Host


Snoop Protocol: FH to MH
9
8
7
Base Station
Sender
ack 6
ack 7
6 1 5 4 3 2
Active soft state agent at base station
Transport-aware reliable link protocol
Preserves end-to-end semantics
Mobile Host

Snoop Protocol: MH to FH

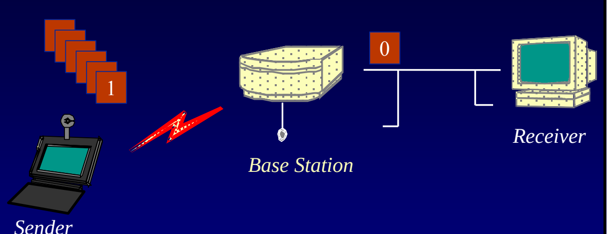


Caching and retransmission will not work
- Losses occur before packet reaches BS
- Losses should not be hidden

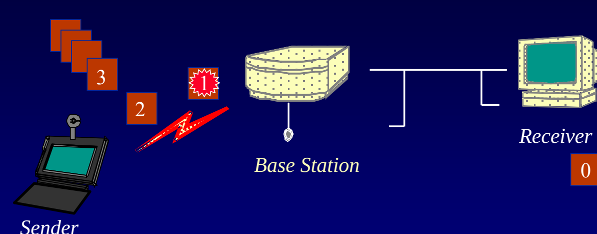
Snoop Protocol: MH to FH

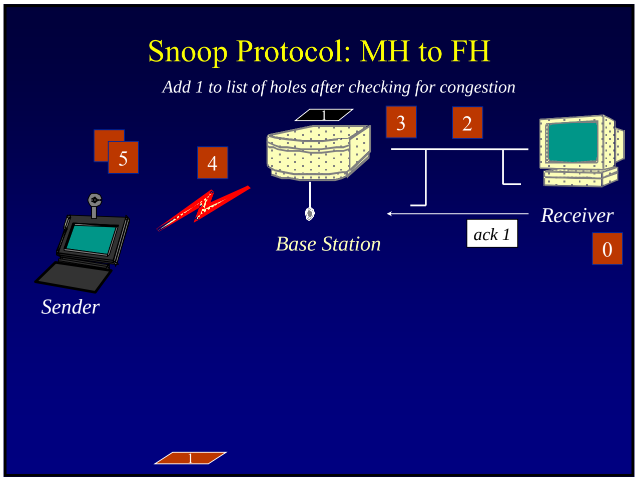
- Solution #1: Negative ACKs (NACKs)
  - NACK from BS to MH on wireless loss
- Solution #2: *Explicit Loss Notifications (ELN)*
  - In-band message to TCP sender
  - General solution framework

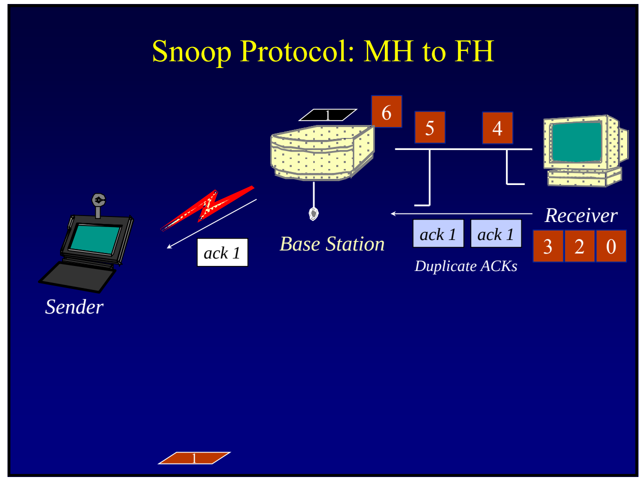
Snoop Protocol: MH to FH

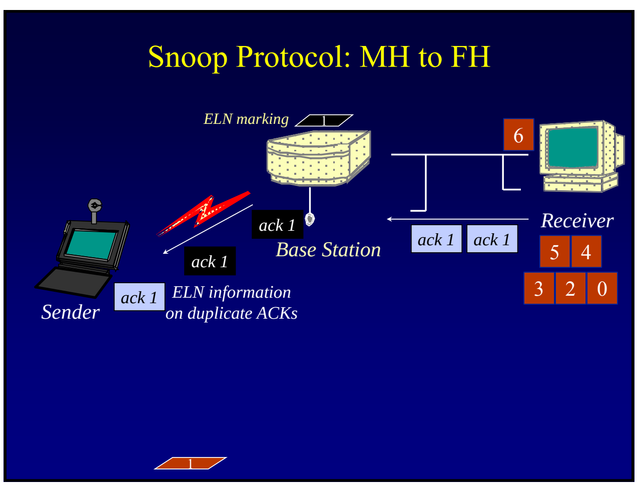


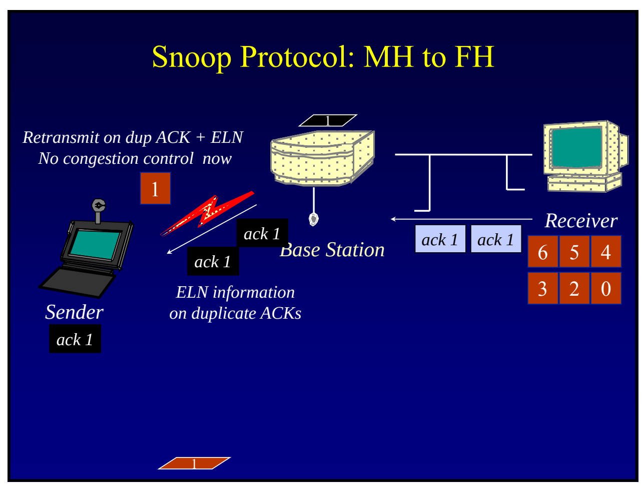
Snoop Protocol: MH to FH



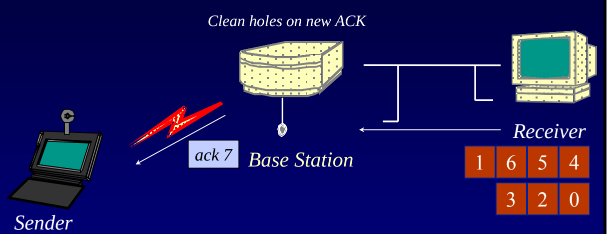

Snoop Protocol: MH to FH
Add 1 to list of holes after checking for congestion
5
4
3
2
ack 1
0
Sender
Base Station
Receiver


Snoop Protocol: MH to FH
6
5
4
ack 1
ack 1
ack 1
Duplicate ACKs
3
2
0
Sender
Base Station
Receiver


Snoop Protocol: MH to FH
ELN marking
6
ack 1
ack 1
ack 1
ack 1
ack 1
ELN information
on duplicate ACKs
5
4
3
2
0
Sender
Base Station
Receiver

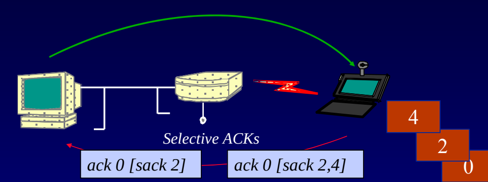

Snoop Protocol: MH to FH
Retransmit on dup ACK + ELN
No congestion control now
1
ack 1
ack 1
ack 1
ack 1
ack 1
ELN information
on duplicate ACKs
6
5
4
3
2
0
Sender
Base Station
Receiver

Snoop Protocol: MH to FH



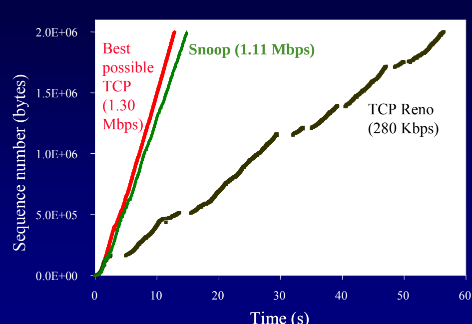
Link-aware transport decouples congestion control from loss recovery. Technique generalizes nicely to wireless transit links

End-to-End Enhancements



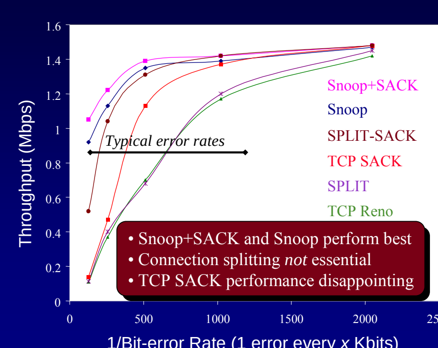
- Decouple congestion control from loss recovery
  - *Explicit Loss Notification (ELN)*
- Burst losses
  - *Selective ACKs (SACKs) [FF96,KM96,MMFR96,B96]*
- Snoop protocol: no changes to fixed hosts on the Internet

Snoop Performance Improvement



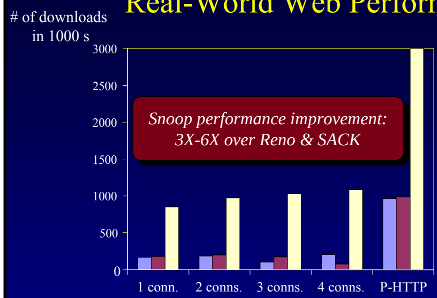
2 MB wide-area TCP transfer over 2 Mbps Lucent WaveLAN

Performance: FH to MH



2 MB local-area TCP transfer over 2 Mbps Lucent WaveLAN

Real-World Web Performance



	1 conn.	2 conns.	3 conns.	4 conns.	P-HTTP
Reno	170	186	102	206	966
SACK	179	203	177	76	985
Snoop	849	975	1033	1085	3000

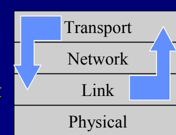
Reno SACK Snoop

Empirical wireless error model from real traces of Reinas wireless network UC Santa Cruz

Empirical Web workload model from real traces

Summary: Wireless Bit-Errors

- Problem: Wireless corruption mistaken for congestion
- Solution: Snoop Protocol
- General lessons
  - *Lightweight soft-state agent in network infrastructure*
    - Fully conforms to the IP service model
    - Automatic instantiation and cleanup
  - *Cross-layer protocol design & optimizations*



Transport-aware link (Snoop agent at BS)

Link-aware transport (ELN)

Snoop Protocol: Disadvantages

- Link layer at base station needs to be TCP-aware
- Not useful if TCP headers are encrypted (IPsec)
- Cannot be used if TCP data and TCP ACKs traverse different paths
  - Both do not go through the same base station, e.g., satellite links

Other Problems in Wireless Networks

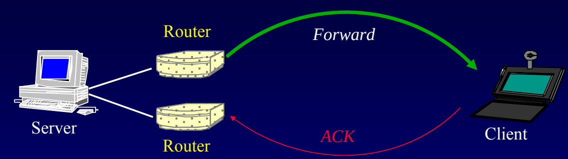
- (#1) Burst errors due to poor signal strength or mobility (handoff)
  - More than one packet lost in TCP window
- (#2) Asymmetric effects
  - Bandwidth asymmetry & latency variability
- (#3) Low channel bandwidth, low throughput
- (#4) Delay is often very high
  - RTT quite long (tunneling, satellite)
  - True in telephone networks providing data services that deploy fixed gateways (non-optimal routes)

Challenge #2: Asymmetric Effects

- Asymmetric access technologies
  - ADSL, (wireless) cable modems, DBS, etc.
  - Low-bandwidth ACK channel [LM97, KVR98]
- Packet radio networks
  - Metricom's Ricochet, CDPD, etc.
  - Adverse interactions between data and ACK flow

Problem: Imperfect ACK feedback degrades TCP performance

The Character of Asymmetry



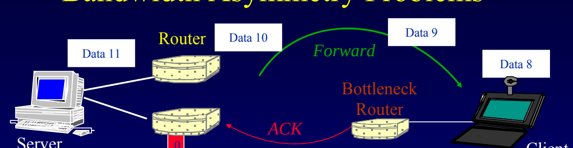
The network and traffic characteristics in one direction significantly affect performance in the other

Bandwidth: 10-1000 times more in the forward direction

Latency: Variability due to MAC protocol interactions

Packet loss: Higher loss- or error-rate in one direction

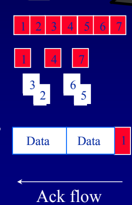
Bandwidth Asymmetry Problems



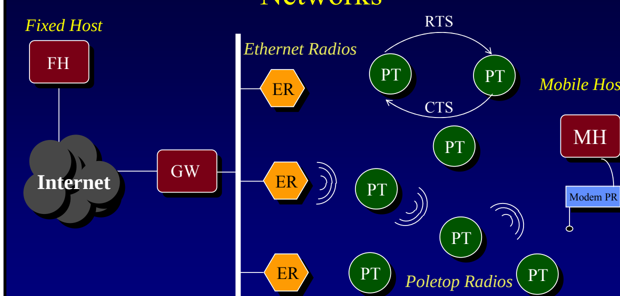
1. Acks arrive slowly (large buffer)

2. Acks are dropped (small buffer)

3. Acks are queued behind data packets



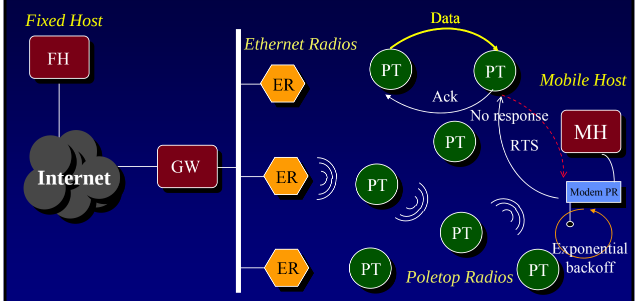
Latency Asymmetry: Packet Radio Networks



Half-duplex radios

Synchronization before communication

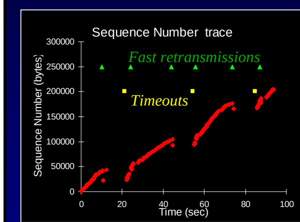
Packet Radio Networks

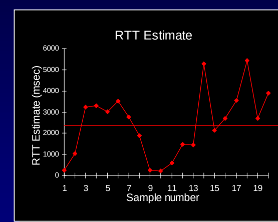


Problem: Large and variable communication latency

Problem: Large Round-Trip Time Variations

Example: Metricom Ricochet Wireless Network





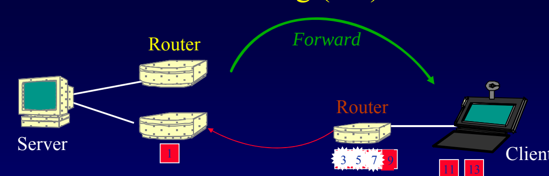
- Mean rtt = 2.45s, std deviation = 1.5s ⇒ long timeout!
- Long idle periods after multiple losses (~ 20 Kbps)
- In contrast, UDP throughput = 50-64 Kbps
- ACK flow affects data latency

Solutions

- Problems arise because of imperfections in the ACK feedback
- Reduce frequency of acks
  - *ACK Filtering (AF)*
  - *ACK Congestion Control (ACC)*
- Handle infrequent acks
  - *Sender Adaptation (SA)*
  - *ACK Reconstruction (AR)*

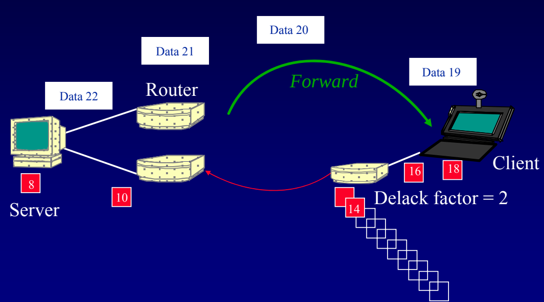
General solution approach for asymmetric situations

ACK Filtering (AF)



- Purge all redundant, cumulative ACKs from constrained reverse queue
- Used in conjunction with sender adaptation or ACK reconstruction

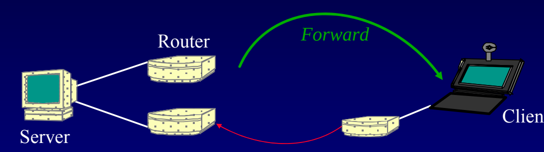
ACK Congestion Control (ACC)



Adaptive extension of TCP delayed ACKs based on congestion feedback from router or sender

Sender Adaptation (SA)

- Infrequent ACKs cause slow window growth
- Sender tends to be bursty



1. cwnd += 8

cwnd += 8/cwnd

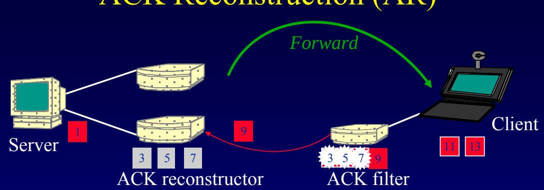
Increment window by *amount* of data ack'd

2. 19 20 21 22 . . .

Regulation: pace packets out at rate estimated by *cwnd/srtt*

This reduces burstiness

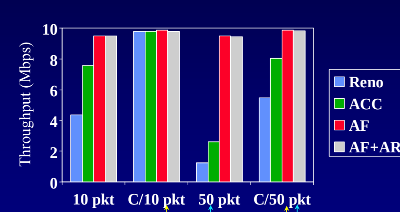
ACK Reconstruction (AR)



- Regenerates ACKs at other end of reverse channel
- Shields sender from large gaps in ack sequence
- AR rate determined by
  - input ACK rate
  - target ACK spacing

Bandwidth Asymmetry Performance

- TCP transfers in the forward direction alone
- Maximum window size 100 KB; no losses on forward path

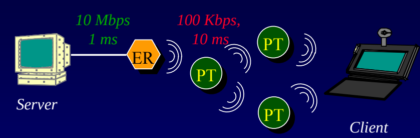


- Header compression helps
- Large reverse channel buffer hurts for Reno and ACC
- Fairness greatly improves using AF and ACC for multiple transfers

Summary: Asymmetric Effects

- *General* definition of asymmetry
  - Problem: ACK channel impacts TCP performance
- *Classification* of types of asymmetry
  - *Bandwidth* asymmetry due to technologies
  - *Latency* asymmetry due to MAC interactions
- *General solutions:* Two-pronged approach
  - Reduce frequency of ACKs (AF, ACC)
  - Handle infrequent ACKs (SA, AR)
- Status
  - BSD/OS 3.0 implementation
  - Soon-to-be Internet RFC

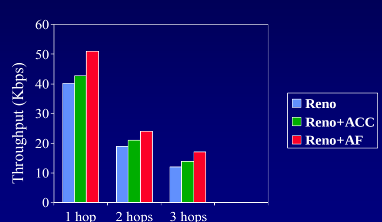
Multihop Wireless Simulations



- 1 to 3 wireless hops on path
- Radio turnaround time = 3-12 ms
- Radio queue size = 10 packets
- Exponential backoff in multiples of 20 ms slots

Performance: Single Transfer

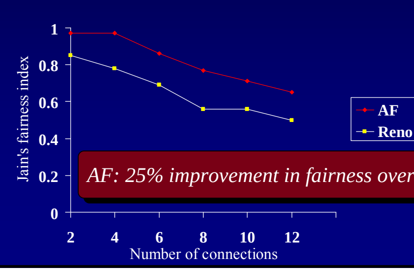
- AF reduces chances that peer radio is busy
  - MAC backoffs less frequent
- Round-trip std deviation reduces from 1.5 s to 0.6 s



AF: 20-35% throughput improvement compared to Reno

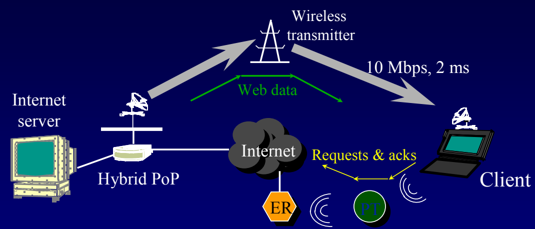
Performance: Concurrent Transfers

- Metrics: *utilization* and *fairness*
- Simultaneous connections over 2-hop network
  - Performance more predictable and consistent with AF
- Unpredictable performance caused by long timeouts



AF: 25% improvement in fairness over Reno

Combining Technologies



Wireless cable forward channel with packet radio reverse channel

Workload: Multiple concurrent Web-like transfers

Issues: both bandwidth and latency asymmetries

Main result: Ack filtering tremendously improves scaling behavior (average completion time vs. # of concurrent transactions)