Mobile Routing

- Mobile IP
- Ad-hoc network routing
- TCP on wireless links
- Wireless MAC
- Assigned reading
  - [BMJ+98] Performance Comparison of Multi-Hop Wireless Ad Hoc Routing Protocols
  - [BPSK97] A Comparison of Mechanism for Improving TCP Performance over Wireless Links
  - [LBC+01] Capacity of Ad Hoc Wireless Networks
  - [She98] A Channel Access Scheme for Large Dense Packet Radio Networks

Wireless Challenges

- Force us to rethink many assumptions
- Need to share airwaves rather than wire
  - Don’t know what hosts are involved
  - Host may not be using same link technology
- Mobility
- Other characteristics of wireless
  - Noisy → lots of losses
  - Slow
  - Interaction of multiple transmitters at receiver
    - Collisions, capture, interference
    - Multipath interference

Cellular Reuse

- Transmissions decay over distance
  - Spectrum can be reused in different areas
  - Different “LANs”
  - Decay is $1/R^2$ in free space, $1/R^4$ in some situations
Overview

- Wireless TCP
- Internet routing
- Ad hoc routing
- Geographic ad hoc routing, MAC, Capacity (Brad Karp)

Wireless Challenges

- Force us to rethink many assumptions
- Need to share airwaves rather than wire
  - Don’t know what hosts are involved
  - Host may not be using same link technology
- Mobility
- Other characteristics of wireless
  - Noisy → lots of losses
  - Slow
  - Interaction of multiple transmitters at receiver
    - Collisions, capture, interference
    - Multipath interference

TCP Problems Over Noisy Links

- Wireless links are inherently error-prone
  - Fades, interference, attenuation
  - Errors often happen in bursts
- TCP cannot distinguish between corruption and congestion
  - TCP unnecessarily reduces window, resulting in low throughput and high latency
- Burst losses often result in timeouts
- Sender retransmission is the only option
  - Inefficient use of bandwidth

Constraints & Requirements

- Incremental deployment
  - Solution should not require modifications to fixed hosts
  - If possible, avoid modifying mobile hosts
- Probably more data to mobile than from mobile
  - Attempt to solve this first
**Challenge #1: Wireless Bit-Errors**

Burst losses lead to coarse-grained timeouts
Result: Low throughput

**Performance Degradation**

Best possible TCP with no errors (1.30 Mbps)
TCP Reno (280 Kbps)

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

**Proposed Solutions**

- End-to-end protocols
  - Selective ACKs, Explicit loss notification
- Split-connection protocols
  - Separate connections for wired path and wireless hop
- Reliable link-layer protocols
  - Error-correcting codes
  - Local retransmission

**Approach Styles (End-to-End)**

- Improve TCP implementations
  - Not incrementally deployable
  - Improve loss recovery (SACK, NewReno)
  - Help it identify congestion (ELN, ECN)
    - ACKs include flag indicating wireless loss
    - Trick TCP into doing right thing → E.g. send extra dupacks
  - What is SMART?
    - DUPACK includes sequence of data packet that triggered it
Approach Styles (Split Connection)

- Split connections
  - Wireless connection need not be TCP
  - Hard state at base station
    - Complicates mobility
    - Vulnerable to failures
    - Violates end-to-end semantics

Split-Connection Congestion Window

- Wired connection does not shrink congestion window
- But wireless connection times out often, causing sender to stall

Approach Styles (Link Layer)

- More aggressive local retransmit than TCP
  - Bandwidth not wasted on wired links
  - Adverse interactions with transport layer
    - Timer interactions
    - Interactions with fast retransmissions
    - Large end-to-end round-trip time variation
  - FEC does not work well with burst losses

Hybrid Approach: Snoop Protocol

- Shield TCP sender from wireless vagaries
  - Eliminate adverse interactions between protocol layers
  - Congestion control only when congestion occurs
- The End-to-End Argument [SRC84]
  - Preserve TCP/IP service model: end-to-end semantics
  - Is connection splitting fundamentally important?
- Eliminate non-TCP protocol messages
  - Is link-layer messaging fundamentally important?

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

- Modify base station
  - to cache un-acked TCP packets
  - … and perform local retransmissions
- Key ideas
  - No transport level code in base station
  - When node moves to different base station, state eventually recreated there

Snoop Protocol: CH to MH

- Snoop agent: active interposition agent
  - Snoops on TCP segments and ACKs
  - Detects losses by duplicate ACKs and timers
  - Suppresses duplicate ACKs from FH sender

- Transfer of file from CH to MH
  - Current window = 6 packets

- Transfer begins
Snoop Protocol: CH to MH

- Snoop agent caches segments that pass by

- Packet 1 is Lost
  - Duplicate ACKs generated
  - Packet 1 retransmitted from cache at higher priority
- Duplicate ACKs suppressed

- Clean cache on new ACK

- Clean cache on new ACK

- Active soft state agent at base station
- Transport-aware reliable link protocol
- Preserves end-to-end semantics
Overview

- Wireless TCP
- Internet routing
- Ad hoc routing

How to Handle Mobile Nodes?

- Dynamic Host Configuration (DHCP)
  - Host gets new IP address in new locations
  - Problems
    - Host does not have constant name/address → how do others contact host?
    - What happens to active transport connections?

- Naming
  - Use DHCP and update name-address mapping whenever host changes address
  - Fixes contact problem but not broken transport connections

Handling Mobile Nodes (Transport)

- TCP currently uses 4 tuple to describe connection
  - \(<\text{Src Addr}, \text{Src port}, \text{Dst addr}, \text{Dst port}>\)
- Modify TCP to allow peer’s address to be changed during connection
- Security issues
  - Can someone easily hijack connection?
- Difficult deployment → both ends must support mobility
Handling Mobile Node

- Link layer mobility
  - Learning bridges can handle mobility \(\rightarrow\) this is how it is handled at CMU
  - Encapsulated PPP (PPTP) \(\rightarrow\) Have mobile host act like he is connected to original LAN
    - Works for IP AND other network protocols
- Multicast
  - Solves similar problem \(\rightarrow\) how to route packets to different sets of hosts at different times
  - Can’t we just reuse same solutions?
    - Don’t really have solution for multicast either!

Handling Mobile Nodes (Routing)

- Allow mobile node to keep same address and name
- How do we deliver IP packets when the endpoint moves?
  - Why can’t we just have nodes advertise route to their address?
- What about packets from the mobile host?
  - Routing not a problem
  - What source address on packet?
- Key design considerations
  - Scale
  - Incremental deployment

Basic Solution to Mobile Routing

- Same as other problems in Computer Science
  - Add a level of indirection
- Keep some part of the network informed about current location
  - Need technique to route packets through this location (interception)
- Need to forward packets from this location to mobile host (delivery)

Interception

- Somewhere along normal forwarding path
  - At source
  - Any router along path
  - Router to home network
  - Machine on home network (masquerading as mobile host)
- Clever tricks to force packet to particular destination
  - "Mobile subnet" – assign mobiles a special address range and have special node advertise route
Delivery

- Need to get packet to mobile’s current location
- Tunnels
  - Tunnel endpoint = current location
  - Tunnel contents = original packets
- Source routing
  - Loose source route through mobile current location
- Network address translation (NAT)
  - What about packets from the mobile host?

Mobile IP (RFC 2290)

- Interception
  - Typically home agent – hosts on home network
- Delivery
  - Typically IP-in-IP tunneling
  - Endpoint – either temporary mobile address or foreign agent
- Terminology
  - Mobile host (MH), correspondent host (CH), home agent (HA), foreign agent (FA)
  - Care-of-address, home address

Mobile IP (MH at Home)

Mobile IP (MH Moving)
Other Mobile IP Issues

- Route optimality
  - Triangle routing
  - Can be improved with route optimization
    - Unsolicited binding cache update to sender
- Authentication
  - Registration messages
  - Binding cache updates
- Must send updates across network
- Handoffs can be slow
- Problems with basic solution
  - Reverse path check for security
  - Do we really need it…

Overview

- Wireless TCP
- Internet routing
- Ad hoc routing
Ad Hoc Routing

- Create multi-hop connectivity among set of wireless, possibly moving, nodes
- Mobile, wireless hosts act as forwarding nodes as well as end systems
- Need routing protocol to find multi-hop paths
  - Needs to be dynamic to adapt to new routes, movement
  - Interesting challenges related to interference and power limitations
  - Low consumption of memory, bandwidth, power
  - Scalable with numbers of nodes
  - Localized effects of link failure

Problems Using DV or LS

- DV protocols may form loops
  - Very wasteful in wireless: bandwidth, power
  - Loop avoidance sometimes complex
- LS protocols: high storage and communication overhead
- More links in wireless (e.g., clusters) - may be redundant → higher protocol overhead

Problems Using DV or LS

- Periodic updates waste power
  - Tx sends portion of battery power into air
  - Reception requires less power, but periodic updates prevent mobile from “sleeping”
- Convergence may be slower in conventional networks but must be fast in ad-hoc networks and be done without frequent updates

Proposed Protocols

- Destination-Sequenced Distance Vector (DSDV)
  - DV protocol, destinations advertise sequence number to avoid loops, not on demand
- Temporally-Ordered Routing Algorithm (TORA)
  - On demand creation of hhh routes based on link-reversal
- Dynamic Source Routing (DSR)
  - On demand source route discovery
- Ad Hoc On-Demand Distance Vector (AODV)
  - Combination of DSR and DSDV: on demand route discovery with hhh routing
**DSR Concepts**

- **Source routing**
  - No need to maintain up-to-date info at intermediate nodes
- **On-demand route discovery**
  - No need for periodic route advertisements

**DSR Components**

- **Route discovery**
  - The mechanism by which a sending node obtains a route to destination
- **Route maintenance**
  - The mechanism by which a sending node detects that the network topology has changed and its route to destination is no longer valid

---

**DSR Route Discovery**

- **Route discovery - basic idea**
  - **Source** broadcasts route-request to **Destination**
  - Each node forwards request by adding own address and re-broadcasting
  - Requests propagate outward until:
    - Target is found, or
    - A node that has a route to Destination is found

---

**C Broadcasts Route Request to F**

Route Request
Source C
Destination F

- A
- B
- C
- D
- E
- G
- H
- Source C
- Destination F
C Broadcasts Route Request to F

H Responds to Route Request

C Transmits a Packet to F

Forwarding Route Requests

- A request is forwarded if:
  - Node is not the destination
  - Node not already listed in recorded source route
  - Node has not seen request with same sequence number
  - IP TTL field may be used to limit scope
- Destination copies route into a Route-reply packet and sends it back to **Source**
Route Cache
- All source routes learned by a node are kept in Route Cache
- Reduces cost of route discovery
- If intermediate node receives RR for destination and has entry for destination in route cache, it responds to RR and does not propagate RR further
- Nodes overhearing RR/RP may insert routes in cache

Sending Data
- Check cache for route to destination
- If route exists then
  - If reachable in one hop
    - Send packet
  - Else insert routing header to destination and send
- If route does not exist, buffer packet and initiate route discovery

Discussion
- Source routing is good for on demand routes instead of a priori distribution
- Route discovery protocol used to obtain routes on demand
  - Caching used to minimize use of discovery
  - Periodic messages avoided
  - But need to buffer packets

Snoop ACK Processing
- Ack arrives (from mobile host)
  - New ack? Yes
    - Yes: Free buffers, Update RTT estimate
    - No: Propagate ack to sender
  - No: Spurious ack
    - Yes: Discard
    - No: Dup ack?
      - Yes: Discard
      - No: Threshold?
        - Yes: Retransmit lost packet
        - No: Next pkt lost

Common case
**Snoop Data Processing**

Packet arrives

- New pkt?
  - No: Forward pkt
  - Yes: Reset local retransmit counter

- New pkt?
  - No: Mark as cong. loss
  - Yes: Cache packet

- In-sequence?
  - Yes: Forward to mobile
  - No: Cache packet

**Benefits of TCP-Awareness**

- 30-35% improvement for Snoop: LL congestion window is small (but no coarse timeouts occur)
- Connection bandwidth-delay product = 25 KB
- Suppressing duplicate acknowledgments and TCP-awareness leads to better utilization of link bandwidth and performance

**Performance: FH to MH**

- Snoop+SACK and Snoop perform best
- Connection splitting not essential
- TCP SACK performance disappointing

**Next Lecture**

- PROJECT CHECKPOINT
  - Related work
  - Preliminary results
  - Detailed timeline
- Multicast
  - routing, reliability, congestion control
- Network measurements
- Readings
  - [DG90] Multicast Routing in Datagram Internetworks and Extended LANs
  - [CRS201] Enabling Conferencing Applications on the Internet using an Overlay Multicast Architecture
  - [LAWD04] A First-Principles Approach to Understanding the Internet's Router-level Topology
  - [Pax97] End-to-End Internet Packet Dynamics
  - [JD04] Ten Fallacies and Pitfalls on End-to-End Available Bandwidth Estimation
  - [Pax04] Strategies for Sound Internet Measurement
Other Issues

• What about mobility?
• What about mobile-to-fixed communication?

Handling Mobility

Send packets to multiple base stations

Resend missed packets from Snoop cache on handoff

Handing Mobility

Snoop Protocol: MH to CH

• Caching and retransmission will not work
  • Losses occur before packet reaches BS
  • Congestion losses should not be hidden
• Solution: Explicit Loss Notifications (ELN)
  • In-band message to TCP sender
- MH begins transfer to CH

- Packet 1 lost on wireless link

- Add 1 to list of holes after checking for congestion

- Duplicate ACKs sent
- ELN information added to duplicate ACKs

- ELN information on duplicate ACKs
- Retransmit on Packet 1 on dup ACK + ELN
- No congestion control now

- Clean holes on new ACK
- Link-aware transport decouples congestion control from loss recovery
- Technique generalizes nicely to wireless transit links

- Header Compression
Low Bandwidth Links

- Efficiency for interactive
  - 40-byte headers vs payload size – 1 byte payload for telnet
- Header compression
  - What fields change between packets?
  - 3 types – fixed, random, differential

TCP Header

<table>
<thead>
<tr>
<th>Source port</th>
<th>Destination port</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequence number</td>
<td></td>
</tr>
<tr>
<td>Acknowledgement</td>
<td></td>
</tr>
<tr>
<td>HdrLen</td>
<td>Flags</td>
</tr>
<tr>
<td>Checksum</td>
<td>Urgent pointer</td>
</tr>
<tr>
<td>Options (variable)</td>
<td>Data</td>
</tr>
</tbody>
</table>

Header Compression

- What happens if packets are lost or corrupted?
  - Packets created with incorrect fields
  - Checksum makes it possible to identify
  - How is this state recovered from?
- TCP retransmissions are sent with complete headers
  - Large performance penalty – must take a timeout, no data-driven loss recovery
  - How do you handle other protocols?

Non-reliable Protocols

- IPv6 and other protocols are adding large headers
  - However, these protocols don’t have loss recovery
  - How to recovery compression state
- Decaying refresh of compression state
  - Suppose compression state is installed by packet X
  - Send full state with X+2, X+4, X+8 until next state
  - Prevents large number of packets being corrupted
- Heuristics to correct packet
  - Apply differencing fields multiple times
- Do we need to define new formats for each protocol?
  - Not really – can define packet description language [mobicom99]
CSMA/CD Does Not Work

- Carrier sense problems
  - Relevant contention at the receiver, not sender
  - Hidden terminal
  - Exposed terminal
- Collision detection problems
  - Hard to build a radio that can transmit and receive at same time

RTS/CTS Approach

- Before sending data, send Ready-to-Send (RTS)
- Target responds with Clear-to-Send (CTS)
- Others who hear defer transmission
  - Packet length in RTS and CTS messages
- If CTS is not heard, or RTS collides
  - Retransmit RTS after binary exponential backoff