Mobile Routing

CSE 561 Lecture 11, Spring 2002.
David Wetherall

The Mobility Problem: What happens if hosts move?

- Implicit assumption that Internet hosts are fixed
  - IP addresses used to name hosts; cached by higher layers
  - IP routing breaks if addresses change location. **Why?**
- Unfortunately, the buying public likes mobility

Problems

- How does a mobile host get a local IP address?
- How do you know which IP address to use when sending to a mobile host?
- If a host moves during communication how do you know how to migrate state to the new IP address?
- Backwards compatibility

Mobile IP: Johnson96

- **Current IETF proposed standard for mobility**
  - Dates back to research in the early 90s
  - IPv4 (RFC 2002), IPv6 version is roughly the same
- **Design constraints**
  - Network layer solution
  - Only requires changes to mobile hosts
  - Stationary hosts oblivious to mobility
  - Incrementally deployable

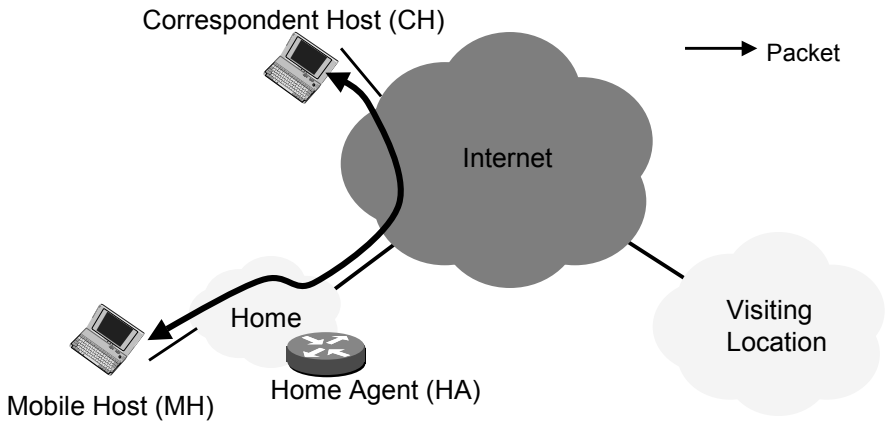
Mobile IP Approach

- **Mobile Host (MH) has two addresses**
- **Home address**
  - Never changes, uniquely identifies the host
  - In "home network"
  - Correspondent host (CH) addresses all packets to the home address
- **Care-of address**
  - Will change, perhaps frequently
  - In "foreign network"
  - Related to current location (IP routing gets it to the right place)

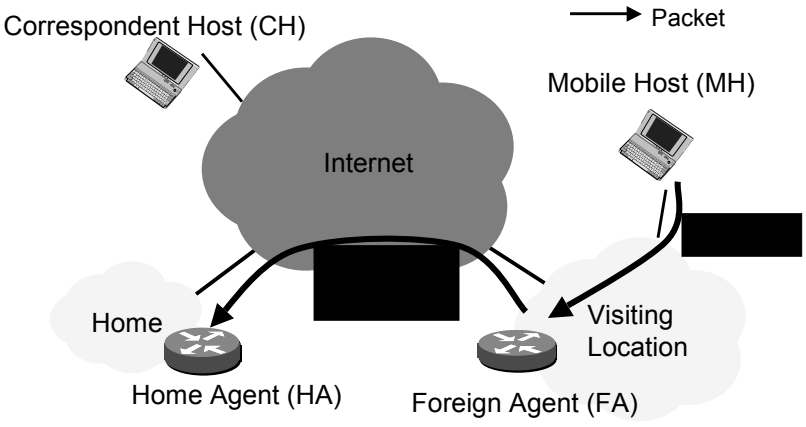
Home and Foreign Agents

- **Home agent** (HA) implements level of indirection between the mobile host and correspondents
  - Accepts traffic sent to home address
    - What about requests ***from*** home network?
  - Tunnels traffic to the mobile host (using care-of address)
  - And vice versa, correspondent none the wiser
- **Foreign agent** (FA) represents mobile in foreign network
  - Foreign agent can be care-of address
    - Mobile host does not need its own address in foreign network
    - Potential advantage: deal with local mobility locally

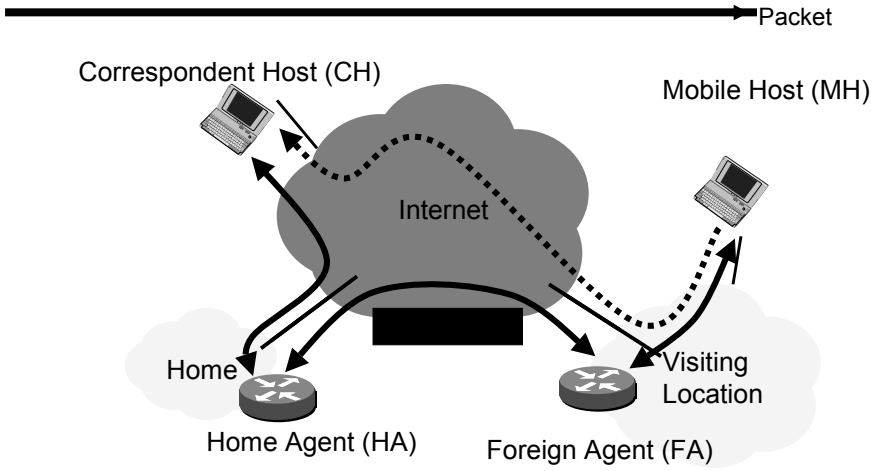
Mobile IP (MH at Home)



Mobile IP (MH Moving)



Mobile IP (MH Away)



Mobile IP Issues

- To make all this happen, a number of issues have to be addressed
  - Discovering agents
  - Registering addresses with agents (establishing bindings)
  - Authentication
  - Tunneling
  - Performance (!)

Agent Discovery

- **Agent discovery** enables a mobile host
  - To notice when it changes networks
  - To notice when it is home again
    - When home, take down the tunnel
  - To find a foreign agent to register with
- Agents multicast **agent advertisements** locally
  - Beacons that tell the mobile who it can hear
  - Start in network A, move to network B
  - Lack of A's beacons and presence of B's tells mobile it has switched networks
- Mobile can also multicast an **agent solicitation**
- Why does multicast work here?

Registration

- Mobiles must register care-of addresses with their home agents
  - So that the home agent knows where to tunnel packets
  - Registration needs to be updated when location changes
- Multiple steps
  - Registration requests first go to foreign agent, then to home agent, which replies to foreign agent, which forwards back to the mobile
- Lifetimes
  - Registrations have TTLs
  - Why?

Registration Authentication

- Registration requests can be used by attackers to hijack tunnels from home agent
  - Hey, send all the mobile's traffic to me now
- Need to authenticate that a registration
  - Came from mobile host (authenticity)
  - Has not been altered (integrity)
  - Is not a replay attack (freshness)
- Mechanisms
  - Shared keys (mobile and home are from same admin domain)
  - MD5 digests
  - Nonces or timestamps

Tunneling

- Home agent and mobile communicate using a tunnel
  - IP in IP encapsulation
- Original packet
  - Correspondent address (src) → mobile home address (dest)
  - Gets sent to home agent
- Tunnel packet
  - Encapsulates original packet
  - Home agent (src) → care-of address (dest)
  - Gets sent to foreign agent (or mobile, depending on care-of)
  - Same forward and reverse. Why no short-cut on reverse?
- Asides
  - Bit of overhead (20 byte header for every packet…poor telnet)

Performance

- The good: No overhead in local operation
  - Home agent out of picture, no longer intercepts packets
  - The common case?
- The bad: Significant overhead in remote operation
  - Triangle routing: Packets between two hosts separated by inches can travel 1000s of miles
  - Wide-area effects determine "local" connection performance
  - The uncommon case? Even so, a steep price to pay
- Hence: Route optimization

Route Optimization

- Route optimization shortcuts the triangle
  - Correspondents can learn and use mobile care-of addresses
  - Tunnel packets directly to care-of address, skip home agent
  - Requires changes to correspondents
    - Or to routers: How likely do you think this is?
- Issues
  - Binding cache updates (consistency)
  - Binding update authentication (more trust)
  - Yet more complexity
  - Necessary for scalability?

Motivating a simple solution …

- **Just travel with your laptop and boot it up elsewhere**
  - **Essentially need a way to access the network: DHCP**
- **Dynamic Host Configuration Protocol (DHCP)**
  - Request IP address dynamically (special broadcast address)
  - How do you get contacted at new IP address?
    - One solution: dynamic DNS
  - Authentication issues (who can use the local 802.11?)
  - Pro: Great over longer time scales…
  - Con: What happens during a session?

E2E Host Mobility: Snoeren00

- E2E Host Mobility: Same goals
  - Do not disrupt connections when network address changes
- Different approach
  - Combination of DNS and connection migration
  - Naming + transport (vs. network-layer w/ Mobile IP)
  - Based upon observation of how connections are made from mobile
- Three components
  - Addressing
  - Locating mobile hosts
  - Connection migration

Addressing

- Mobiles obtain an network-local IP address
  - No home agent, no home address
  - No foreign agent
  - No tunneling
  - Communication between correspondent and mobile uses addresses directly
- Problem: How does the correspondent learn the mobile's address?
  - Client case: if the mobile initiates the connection, the mobile tells the correspondent its address with the SYN packet
  - What about mobile servers?

Locating Mobiles

- Observation: Whenever connections are established, a DNS lookup is performed
- Idea: Use the DNS lookup to return latest mobile address to correspondent
  - In Mobile IP, home address is used to unique identify mobile
  - In E2E, DNS name is used for this purpose
  - Force lookups by setting DNS response TTL to 0
    - What are the performance implications?
  - When mobile moves and obtains a new IP address, it updates its DNS entry (using secure DNS)
    - Opportunity for a race condition
    - Proposed solution: Application-level retries

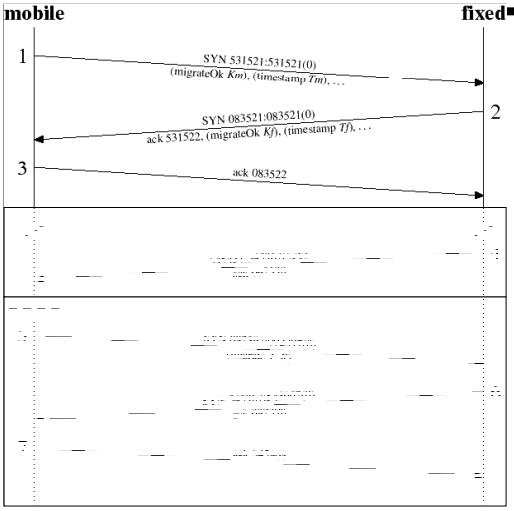
Connection Migration

- Problem: What about existing open connections?
- Solution: TCP Connection Migration
  - New IP Option: Migrate
  - Negotiated with Migrate-Permitted option in SYN
  - Requires modification to TCP stacks at both ends
  - A solution for TCP – what about other transports?

Basic Idea

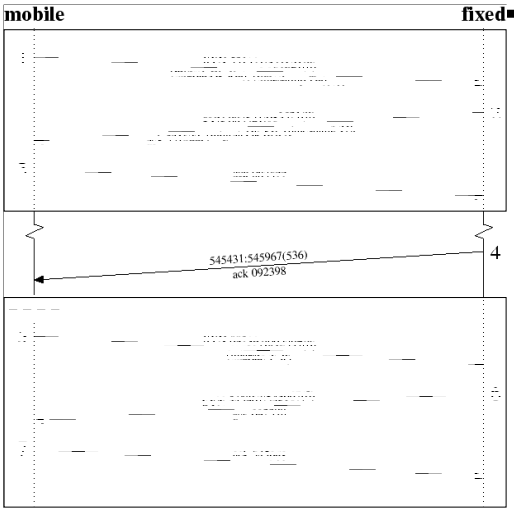
- We have an open connection between correspondent (src) and mobile (dest)
  - Doesn't matter who initiated the connection
  - Connection represented by
    - <src IP, src port, dest IP, dest port>
  - Mobile moves
    - Now has new <dest IP*, dest port*>
  - Want to change connection to
    - <src IP, src port, dest IP*, dest port*>
  - Mobile creates a new connection to the correspondent, forces correspondent to migrate old connection to new one
  - Uses token to show that connections are connected

TCP Migration example



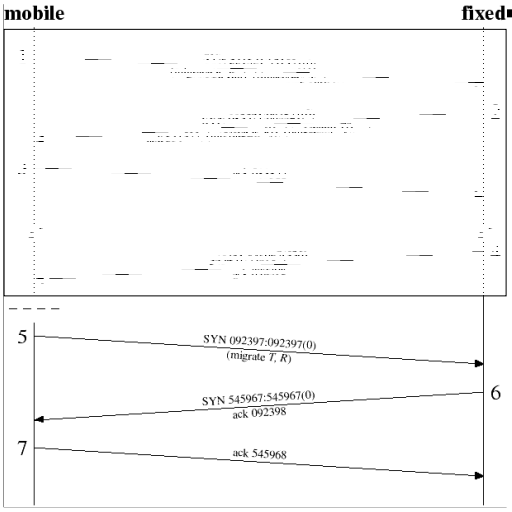
1. Initial SYN
2. SYN/ACK
3. ACK (with data)
4. *Normal data transfer*
5. Migrate SYN
6. Migrate SYN/ACK
7. ACK (with data)

TCP Migration example



1. Initial SYN
2. SYN/ACK
3. ACK (with data)
4. *Normal data transfer*
5. Migrate SYN
6. Migrate SYN/ACK
7. ACK (with data)

TCP Migration example



1. Initial SYN
2. SYN/ACK
3. ACK (with data)
4. *Normal data transfer*
5. Migrate SYN
6. Migrate SYN/ACK
7. ACK (with data)

Issues

- Connection Migration: Devil in the details
  - Migrate-Permitted option: Negotiate migratability
  - Migrate option: Initiate migration
  - MIGRATE_WAIT state: race between reassignment&migration
  - Security: DoS, hijacking (cf. Mobile IP), keys, IPsec
  - Deployment: Have to change all hosts!
  - Other transport protocols: Can easily generalize?
  - No simultaneous migration (i.e. no ad hoc networks)
  - What happens if disconnected for long period (e.g. 5 mins)? Why?
- Pluses
  - No change to routing infrastructure
  - No triangle routes

GPSR (Karp00)

- Ad hoc networks vs. hybrids with fixed infrastructure
  - How realistic? Domain is still being refined.
- Approaches to make routing scale:
  - Hierarchy – but not well suited to mobility
  - On-demand routes – relate to forwarding needs
  - Geography – this paper is about it work
- GPSR: greedy forwarding isn't enough, add RHR.
  - Simple idea, planarity complications
  - The location service is a large missing component