Supporting Mobility

- Need an indirection mechanism that maps a location independent name to current location
- How should this be implemented?
  - Invisibly by the network
  - Some options to do that that people have investigated
    - Network layer solution (e.g., MobileIP)
    - End to end (e.g., using DNS in transport protocol–Snorenen’s paper)
    - Architect an indirection layer with a different application level name space (HIP)
    - via proxy (M-socks)
    - Publish subscribe through an overlay do the mapping (III)

Routing Layer Solution – Problem

- Host identifier (e.g., IP address) is topologically meaningful – Binghamton computers 128.226.x.x, MIT 18.x.x.x
- Same situation as telephones – cant receive calls made to 607 777 2000 when I move to Cambridge, MA
- Retain host address: Routing fails
- Change host address: Lost connections

New Topic – Supporting Mobility

- Would like – be able to freely roam and not be aware of it from a connectivity perspective
  - You can move in your car while talking on a cellular phone
  - Logical connection should be maintained seemlessly as the physical connection changes
- Problem: Naming and Addressing coupled on the Internet
  - Internet was developed for unicast communication between two stationary hosts
  - IP address serves as both a name and an address
  - With mobility, would like name to be decoupled from address
Mobile IP

- Allows a host to be reachable at the same address even as it moves across different networks
- Makes it seem like one network extends over the entire internet continuous connectivity: seamless roaming (application continues running even as the node moves)
- Location independent access to computing resources
- Many initial efforts

Routing table at Network A Gateway (a.b.c.d) is a gateway one hop away from G1

<table>
<thead>
<tr>
<th>Dest</th>
<th>Gateway</th>
<th>Mask</th>
<th>Iface</th>
</tr>
</thead>
<tbody>
<tr>
<td>128.226.3.0</td>
<td>128.226.3.1</td>
<td>255.255.255.0</td>
<td>eth0</td>
</tr>
<tr>
<td>127.0.0.0</td>
<td>127.0.0.1</td>
<td>255.0.0.0</td>
<td>lo</td>
</tr>
<tr>
<td>0.0.0.0</td>
<td>a.b.c.d</td>
<td>0.0.0.0</td>
<td>eth1</td>
</tr>
</tbody>
</table>

Routing table on 128.226.3.30

<table>
<thead>
<tr>
<th>Dest</th>
<th>Gateway</th>
<th>Mask</th>
<th>Iface</th>
</tr>
</thead>
<tbody>
<tr>
<td>128.226.3.28</td>
<td>128.226.3.30</td>
<td>255.255.255.255</td>
<td>eth0*</td>
</tr>
<tr>
<td>128.226.3.0</td>
<td>128.226.3.30</td>
<td>255.255.255.0</td>
<td>eth0</td>
</tr>
<tr>
<td>127.0.0.0</td>
<td>127.0.0.1</td>
<td>255.0.0.0</td>
<td>lo</td>
</tr>
<tr>
<td>0.0.0.0</td>
<td>128.226.3.1</td>
<td>0.0.0.0</td>
<td>eth0</td>
</tr>
</tbody>
</table>

Key idea: Encapsulation/Tunneling

- When the Mobile Host (MH) is away, its Home Agent (HA) picks up its datagrams, encapsulates them in a new datagram and routes the new datagram in the normal way
- Destination field in the outer IP header is a “care-of” address on the foreign network
- Intermediate routers are unaware of the inner IP header – tunneling

Tunneling Review

- Recall that tunneling allows one IP packet to be encapsulated within a new one
- Just add another IP header to it; protocol number is 4
- Minimal encapsulation available to reduce the overhead of the additional header (protocol number becomes 55)
- Handles fragmented packets
On the Home Network

- The "care of" address used in encapsulation may belong to the FA or may be a temporary address acquired by the MH (e.g., using DHCP).

- The host used as the tunnel endpoint must have kernel support for decapsulation.

- If the Gateway is the HA, picking up MH packets trivial.

- HA pretends to be MH, responds to requests for MH's physical address with its own.

On the Foreign Network

- The "care of" address used in encapsulation may belong to the FA or may be a temporary address acquired by the MH (e.g., using DHCP).

- The host used as the tunnel endpoint must have kernel support for decapsulation.

- If the HA is not the gateway, proxy ARP must be used.

Routing Table at HA

- If the Gateway is the HA, picking up MH packets trivial.

- HA pretends to be MH, responds to requests for MH's physical address with its own.

Implementation

- For encapsulation, could add code in the kernel procedure that receives IP datagrams.

- Inefficient; it is on the critical path of all received IP packets.

- Better solution: create a special software interface (like the loopback) that simply adds a new IP header to incoming datagrams before doing regular routing.

- Create a special software interface that receives IP datagrams for encapsulation, could add code in the kernel.

- Add a route to MH through this interface in the routing table but beware of routing loops.

- Add a route to MH through this interface in the routing table but beware of routing loops.

- Add a route to MH through this interface in the routing table but beware of routing loops.

- Add a route to MH through this interface in the routing table but beware of routing loops.
Foreign Agent Decapsulates

Routing Table on FA
Dest | Gateway | Mask | Iface
--- | --- | --- | ---
128.226.3.30 | 128.6.5.1 | 255.255.255.255 | eth0
128.6.5.0 | 128.6.5.1 | 255.255.255.0 | eth0
127.0.0.0 | 127.0.0.1 | 255.0.0.0 | lo
0.0.0.0 | p.q.r.s | 0.0.0.0 | eth1

Routing Table on MH
Dest | Gateway | Mask | Iface
--- | --- | --- | ---
128.226.3.30 | 128.6.5.1 | 255.255.255.255 | dummy*
128.226.3.30 | 128.226.3.30 | 255.255.255.255 | eth0
127.0.0.0 | 127.0.0.1 | 255.0.0.0 | lo
0.0.0.0 | 128.6.5.1 | 0.0.0.0 | eth0

*Why not use the following?
128.6.5.0 | 128.226.3.30 | 255.255.255.0 | eth0

Mobile Host Decapsulates

Routing Table on FA
Dest | Gateway | Mask | Iface
--- | --- | --- | ---
128.6.5.0 | 128.6.5.1 | 255.255.255.0 | eth0
127.0.0.0 | 127.0.0.1 | 255.0.0.0 | lo
0.0.0.0 | p.q.r.s | 0.0.0.0 | eth1

Routing Table on MH
Dest | Gateway | Mask | Iface
--- | --- | --- | ---
128.226.3.30 | 128.226.3.30 | 255.255.255.255 | dummy*
128.6.5.1 | 128.6.5.15 | 255.255.255.255 | eth0
127.0.0.0 | 127.0.0.1 | 255.0.0.0 | lo
0.0.0.0 | 128.6.5.1 | 0.0.0.0 | eth0

*Another software interface, useful as a place holder for an IP address

What about forwarding of broadcast datagrams?

Registering to a Foreign Network

Router Discovery (RFC 1256): A pair of ICMP messages for router advertisement and solicitation

- Mobility Agents (HAs, FAs) periodically send out advertisements as link-level broadcasts
- Sent as an extension to router advertisement ICMP messages
- Advertisement includes care-of address, encapsulation type and other status information
- The tuple (mobile host, care-of address, last message ID, lifetime) is called a binding. Registration updates binding.
- Authentication needed to prevent connection hijacking (several authentication extensions defined)
- Uses UDP
Discussion/Other Issues

- Issues with mobility during registration
- Moving around Firewalls
- TCP and Mobile IP (we’ll discuss in detail in a few classes)
- Implications on scalability of routing
- Interaction with Ingress filtering
  - Widely deployed “filtering” technique to prevent spoofing/denial of service attacks
  - A packet is dropped if its source address does not match the address of the network it originated from
  - What does this do to mobility? What is the solution?
    * Reverse tunneling; even more triangulation

Alternative Model – End to End Mobility

- What is the motivation? What is wrong with Mobile IP?
  - Hurts routing scalability
  - Inefficient – triangular routing, tunneling
    * Eliminating triangular routing increases deployment barrier
  - Complex – see some of the issues above
  - Many types of applications; may not be appropriate for all

Triangular Route Optimization

- Perkins and Johnson proposed an optimization to eliminate route triangulation
  - Basic Idea: HA securely exchanges the c/o address with a node that is communicating with the mobile
    * Node caches the c/o address locally and uses it to send packets directly to destination
    * Proposed solution also allows intermediate routers to do the same
Nice overview of mobility approaches and their pros and cons organized into:

- Network layer solutions
  - Mobile IP
  - Give mobiles multicast addresses (it is not tied to a location)
  - Translates the problem to a multicast problem
- Higher layer solutions (Transport/Application)
  - MSOCKs:
    - Split the connection into two parts, one to HA and one from HA to mobile
    - When node moves, HA establishes a new connection to it

Proposed Solution

- Recall Domain Naming System (DNS) provides translation from FQDN to IP address (e.g., yahoo.com -> 64.58.79.230)
- Idea: use DNS to provide resolution from Name to location point of attachment
- When a mobile moves, it updates its DNS server to point to its new location
- How to get around DNS replication/caching?

End to End solution

- Leave it to the application to decide what happens when mobility occurs (end to end principle)
- For seamless mobility (i.e., like Mobile IP), need to migrate existing connections
- They present a scheme for migrating TCP connections
  - Adds a migrate option to TCP
  - Exchange a connection token during connection establishment
  - When a node moves, it uses the token to reestablish the connection from its new location

TCP State Diagram
Security Implications

- Connection hijacking can occur if anyone intercepts the token
  - One solution – use IPSEC to encrypt everything
    - Problem: IPSEC isn’t widely deployed

- Proposed solution – use a secure key exchange algorithm (they use Elliptic Curve Diffie Hellman) to generate token
  - Still has a couple of problems

Concurrent Migration can be supported by detecting the case and doing a new DNS resolution through an upcall (not a clean solution)
Discussion

• Comparison to Mobile IP

• NAT and proxies (informal argument that it works well with both)

• Is deployment any easier?
  – It's at the ends, but both ends
  – Need changes for each different protocol that requires mobility support