Chapter 8
MOBILE IP AND TCP
Mobile Computing
Summer 2002

Overview
- Network Protocols / Mobile IP
  - Motivation
  - Data transfer
  - Encapsulation
  - Problems
  - DHCP
- Mobile Transport Layer / TCP
  - Motivation
  - Various TCP mechanisms

Motivation for Mobile IP
- Routing
  - based on IP destination address, network prefix (e.g. 129.132.13) determines physical subnet
  - change of physical subnet implies change of IP address to have a topological correct address (standard IP) or needs special entries in the routing tables
- Changing the IP-address?
  - adjust the host IP address depending on the current location
  - almost impossible to find a mobile system, DNS updates are too slow
  - TCP connections break
  - security problems
- Change/Add routing table entries for mobile hosts?
  - worldwide!
  - does not scale with the number of mobile hosts and frequent changes in their location

Requirements to Mobile IP (RFC 2002)
- Compatibility
  - support of the same layer 2 protocols as IP
  - no changes to current end-systems and routers required
  - mobile end-systems can communicate with fixed systems
- Transparency
  - mobile end-systems keep their IP address
  - continuation of communication after interruption of link possible
  - point of connection to the fixed network can be changed
- Efficiency and scalability
  - only little additional messages to the mobile system required (connection typically via a low bandwidth radio link)
  - world-wide support of a large number of mobile systems
- Security
  - authentication of all registration messages
Data transfer to the mobile system

1. Sender sends to the IP address of MN, HA intercepts packet (proxy ARP)
2. HA tunnels packet to COA, here FA, by encapsulation
3. FA forwards the packet to the MN

Data transfer from the mobile system

1. Sender sends to the IP address of the receiver as usual, FA works as default router

Terminology

- Mobile Node (MN)
  - system (node) that can change the point of connection to the network without changing its IP address
- Home Agent (HA)
  - system in the home network of the MN, typically a router
  - registers the location of the MN, tunnels IP datagrams to the COA
- Foreign Agent (FA)
  - system in the current foreign network of the MN, typically a router
  - typically the default router for the MN
- Care-of Address (COA)
  - address of the current tunnel end-point for the MN (at FA or MN)
  - actual location of the MN from an IP point of view
  - can be chosen, e.g., via DHCP
- Correspondent Node (CN)
Overview

Network integration

- Agent Advertisement
  - HA and FA periodically send advertisement messages into their physical subnets
  - MN listens to these messages and detects, if it is in the home or a foreign network (standard case for home network)
  - MN reads a COA from the FA advertisement messages
- Registration (always limited lifetime!)
  - MN signals COA to the HA via the FA, HA acknowledges via FA to MN
  - these actions have to be secured by authentication
- Advertisement
  - HA advertises the IP address of the MN (as for fixed systems), i.e. standard routing information
  - routers adjust their entries, these are stable for a longer time (HA responsible for a MN over a longer period of time)
  - packets to the MN are sent to the HA,
  - independent of changes in COA/FA

Agent advertisement

Registration

- COA @ FA:
- COA @ MN:
Mobile IP registration request

- **type**
- **home address**
- **home agent**
- **COA**
- **identification**
- extensions

Tunneling and Encapsulation

- **original IP header**
- **original data**
- **new IP header**
- **new data**
- **outer header**
- **inner header**
- **original data**

IP-in-IP Encapsulation

- Mandatory in RFC 2003
- tunnel between HA and COA

Minimal Encapsulation

- optional
- avoids repetition of identical fields such as TTL, IHL, version, TOS
- only applicable for unfragmented packets, no space left for fragment identification
Generic Routing Encapsulation

<table>
<thead>
<tr>
<th>Original header</th>
<th>Original data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Outer header</td>
<td>GRE header</td>
</tr>
<tr>
<td>GRE header</td>
<td>Original header</td>
</tr>
<tr>
<td>New header</td>
<td>New data</td>
</tr>
</tbody>
</table>

Optimization of packet forwarding

- **Triangular Routing**
  - sender sends all packets via HA to MN
  - higher latency and network load

- **"Solutions"**
  - sender learns the current location of MN
  - direct tunneling to this location
  - HA informs a sender about the location of MN
  - big security problems

- **Change of FA**
  - packets on-the-fly during the change can be lost
  - new FA informs old FA to avoid packet loss, old FA now forwards remaining packets to new FA
  - this information also enables the old FA to release resources for the MN

Change of foreign agent

![Diagram of change of foreign agent]

Data transfer from the mobile system

![Diagram of data transfer from the mobile system]

Problems:
- Firewall at CN
- TTL
- Multicast
Reverse tunneling (RFC 2344)

1. MN sends to FA
2. FA tunnels packets to HA by encapsulation
3. HA forwards the packet to the receiver (standard case)

Mobile IP with reverse tunneling

- Router accepts often only “topologically correct” addresses (firewalls)
  - A packet from the MN encapsulated by the FA is now topologically correct
  - Furthermore, multicast and TTL problems solved (TTL in the home network correct, but MN is too far away from the receiver)
- Reverse tunneling does not solve
  - Problems with firewalls, the reverse tunnel can be abused to circumvent security mechanisms (tunnel hijacking)
  - Optimization of data paths, i.e. packets will be forwarded through the tunnel via the HA to a sender (double triangular routing)
- Reverse tunneling is backwards compatible
  - The extensions can be implemented easily and cooperate with current implementations without these extensions

Mobile IP and IPv6

- Mobile IP was developed for IPv4, but IPv6 simplifies the protocols
  - Security is integrated and not an add-on, authentication of registration is included
  - COA can be assigned via auto-configuration (DHCPv6 is one candidate), every node has the address auto-configuration
  - No need for a separate FA, all routers perform router advertisement which can be used instead of the special agent advertisement
  - MN can signal a sender directly the COA, sending via HA not needed in this case (automatic path optimization)
  - "Soft" hand-over, i.e. without packet loss, between two subnets is supported
    - MN sends the new COA to its old router
    - The old router encapsulates all incoming packets for the MN and forwards them to the new COA
    - Authentication is always granted

DHCP: Dynamic Host Configuration Protocol

- Application
  - Simplification of installation and maintenance of networked computers
  - Supplies systems with all necessary information, such as IP address, DNS server address, domain name, subnet mask, default router etc.
  - Enables automatic integration of systems into an Intranet or the Internet, can be used to acquire a COA for Mobile IP
- Client/Server-Model
  - The client sends via a MAC broadcast a request to the DHCP server (might be via a DHCP relay)
DHCP - protocol mechanisms

- Server (not selected)
- Client
- Server (selected)
- Time

- DHCPDISCOVER
- DHCPDISCOVER
- DHCPOFFER
- DHCPOFFER
- DHCPREQUEST
- DHCPREQUEST
- DHCPACK
- DHCPRELEASE
- DHCPRELEASE
- DHCPOFFER
- DHCPREQUEST
- DHCPDISCOVER
- DHCPDISCOVER
- DHCPREQUEST

Determine the configuration

DHCP characteristics

- Server
  - several servers can be configured for DHCP, coordination not yet standardized (i.e., manual configuration)
- Renewal of configurations
  - IP addresses have to be requested periodically, simplified protocol
- Options
  - available for routers, subnet mask, NTP (network time protocol) timeserver, SLP (service location protocol) directory, DNS (domain name system)
- Security problems
  - no authentication of DHCP information specified

TCP Overview

- Transport control protocols typically designed for
  - Fixed end-systems in wired networks
- Research activities
  - Performance
  - Congestion control
  - Efficient retransmissions
- TCP congestion control
  - packet loss in fixed networks typically due to (temporary) overload situations
  - router have to discard packets as soon as the buffers are full
  - TCP recognizes congestion only indirectly via missing acknowledgements, retransmissions unwise, they would only contribute to the congestion and make it even worse

TCP slow-start

- sender calculates a congestion window for a receiver
- start with a congestion window size equal to one segment
- exponential increase* of the congestion window up to the congestion threshold, then linear increase
- missing acknowledgement causes the reduction of the congestion threshold to one half of the current congestion window
- congestion window starts again with one segment

*slow-start vs. exponential increase: window is increased by one for each acknowledgement, that is, 1 → 2 → 4 → 8 … In other words, the slow-start mechanism is rather a “quick-start”.
TCP fast retransmit/fast recovery

- TCP sends an acknowledgement only after receiving a packet
- If a sender receives several acknowledgements for the same packet, this is due to a gap in received packets at the receiver
- Sender can retransmit missing packets (fast retransmit)
- Also, the receiver got all packets up to the gap and is actually receiving packets
- Therefore, packet loss is not due to congestion, continue with current congestion window (fast recovery)
- In the following simplified analysis, we do consider neither fast retransmit nor fast recovery.

TCP on lossy (wireless) link

- Without fast retransmit/fast recovery

Very high loss probability
High loss probability

Simple analysis model for lossy TCP

- Segment loss probability \( q = 1-p \)
- We are interested in the throughput \( T \)
- If there are no losses (and all ACKs are received in time):
  - Number of segments \( S \) first doubles up to threshold \( W \) (slow start phase)
  - Number of segments \( S \) is incremented after \( S \) successful ACKs
  - At some point we reach the bandwidth of the channel \( B \)
- If there is a loss
  - We go back to \( S = 1 \)

Simple analysis of lossy TCP

- For not too high error probability \( q \) we are usually in the congestion mode (that is: not the slow start mode).
- The expected number of successful transmission \( E \) before we get an error is
  \[
  E = \sum_{i=0}^{\infty} i \cdot p^i (1-p) = \frac{p}{1-p}
  \]
- In the equilibrium, we are in the states 1, 2, 3, …, \( S-1, S \), and then back to 1 because we have a missing ACK, that is, we have \( 1+2+\ldots+S = S^2/2 \) successful transmissions.
- With \( S^2/2 = E = p/q \) we get \( T = S/2 = \sqrt{1/2 \cdot p/q} = \Theta(1/\sqrt{q}) \), for \( p = \Theta(1) \)
Lossy TCP: Graphical Interpretation

\[ T(p) = \min \left( B, \sqrt{\frac{p}{2(1-p)}} \right) \]

- Plot of \( T(p) \), with \( B = 50 \)
- Note that 1% faulty transmissions is enough to degrade the throughput to about 14% of the bandwidth.
- 10% error rate gives about 4% of possible bandwidth.
- The higher the bandwidth, the worse the relative loss.

Mobility and TCP

- TCP assumes congestion if packets are dropped
  - typically wrong in wireless networks, here we often have packet loss due to transmission errors
  - furthermore, mobility itself can cause packet loss, if e.g. a mobile node roams from one access point (e.g. foreign agent in Mobile IP) to another while there are still packets in transit to the wrong access point and forwarding is not possible

- The performance of an unchanged TCP degrades severely
  - however, TCP cannot be changed fundamentally due to the large base of installation in the fixed network, TCP for mobility has to remain compatible
  - the basic TCP mechanisms keep the whole Internet together

Indirect TCP (I-TCP)

- segments the connection
  - no changes to the TCP protocol for hosts connected to the wired Internet, millions of computers use (variants of) this protocol
  - optimized TCP protocol for mobile hosts
  - splitting of the TCP connection at, e.g., the foreign agent into two TCP connections, no real end-to-end connection any longer
  - hosts in the fixed part of the net do not notice the characteristics of the wireless part

I-TCP socket and state migration

- socket migration and state transfer
- access point
  - mobile host
  - "wired" Internet
  - "wireless" TCP
  - standard TCP
Indirect TCP Advantages and Disadvantages

+ no changes in the fixed network necessary, no changes for the hosts (TCP protocol) necessary, all current optimizations to TCP still work
+ transmission errors on the wireless link do not propagate into the fixed network
+ simple to control, mobile TCP is used only for one hop, between a foreign agent and a mobile host
+ therefore, a very fast retransmission of packets is possible, the short delay on the mobile hop is known
  – loss of end-to-end semantics, an acknowledgement to a sender does now not any longer mean that a receiver really got a packet, foreign agents might crash
  – higher latency possible due to buffering of data with the foreign agent and forwarding to a new foreign agent
  – high trust at foreign agent; end-to-end encryption impossible

Snooping TCP

• Transparent extension of TCP within the foreign agent
  – buffering of packets sent to the mobile host
  – lost packets on the wireless link (both directions!) will be retransmitted immediately by the mobile host or foreign agent, respectively (so called “local” retransmission)
  – the foreign agent therefore “snoops” the packet flow and recognizes acknowledgements in both directions, it also filters ACKs
  – changes of TCP only within the foreign agent

Snooping TCP

• Data transfer to the mobile host
  – FA buffers data until it receives ACK of the MH, FA detects packet loss via duplicated ACKs or time-out
  – fast retransmission possible, transparent for the fixed network
• Data transfer from the mobile host
  – FA detects packet loss on the wireless link via sequence numbers, FA answers directly with a NACK to the MH
  – MH can now retransmit data with only a very short delay
• Integration of the MAC layer
  – MAC layer often has similar mechanisms to those of TCP
  – thus, the MAC layer can already detect duplicated packets due to retransmissions and discard them
• Problems
  – snooping TCP does not isolate the wireless link as good as I-TCP
  – snooping might be useless depending on encryption schemes

Mobile TCP

• Special handling of lengthy and/or frequent disconnections
• M-TCP splits as I-TCP does
  – unmodified TCP fixed network to supervisory host (SH)
  – optimized TCP SH to MH
• Supervisory host
  – no caching, no retransmission
  – monitors all packets, if disconnection detected
    • set sender window size to 0
    • sender automatically goes into persistent mode
  – old or new SH re-open the window
+ maintains end-to-end semantics, supports disconnection, no buffer forwarding
  – does not solve problem of bad wireless link, only disconnections
  – adapted TCP on wireless link; new software needed
Fast retransmit/fast recovery

- Problem: Change of foreign agent often results in packet loss
  - TCP reacts with slow-start although there is no congestion

- Solution: Forced fast retransmit
  - as soon as the mobile host has registered with a new foreign agent, the MH sends (three) duplicated acknowledgements on purpose
  - this forces the fast retransmit mode at the communication partners
  - additionally, the TCP on the MH is forced to continue sending with the actual window size and not to go into slow-start after registration

  + simple changes result in significant higher performance
  - what a hack…

Transmission/time-out freezing

- Mobile hosts can be disconnected for a longer time
  - no packet exchange possible, e.g., in a tunnel, disconnection due to overloaded cells or multiplex with higher priority traffic
  - TCP disconnects after time-out completely

- TCP freezing
  - MAC layer is often able to detect interruption in advance
  - MAC can inform TCP layer of upcoming loss of connection
  - TCP stops sending, but does now not assume a congested link
  - MAC layer signals again if reconnected

  + scheme is independent of data
  - TCP on mobile host has to be changed, mechanism depends on MAC layer

Selective retransmission

- TCP acknowledgements are often cumulative
  - ACK n acknowledges correct and in-sequence receipt of packets up to n
  - if single packets are missing quite often a whole packet sequence beginning at the gap has to be retransmitted (go-back-n), thus wasting bandwidth, especially if the bandwidth-delay product is high.

- Selective retransmission as one solution
  - RFC2018 allows for acknowledgements of single packets, not only acknowledgements of in-sequence packet streams without gaps
  - sender can now retransmit only the missing packets

  + much higher efficiency
  - more complex software in a receiver, more buffer needed at the receiver

Transaction oriented TCP

- TCP phases
  - connection setup, data transmission, connection release
  - using 3-way-handshake needs 3 packets for setup and release, respectively
  - thus, even short messages need a minimum of 7 packets!

- Transaction oriented TCP
  - RFC1644, T-TCP, describes a TCP version to avoid this overhead
  - connection setup, data transfer and connection release can be combined
  - thus, only 2 or 3 packets are needed

  + Efficiency
  - Requires changed TCP
Comparison of different approaches for a “mobile” TCP

<table>
<thead>
<tr>
<th>Approach</th>
<th>Mechanism</th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Indirect TCP</td>
<td>splits TCP connection into two connections</td>
<td>isolation of wireless link, simple</td>
<td>loss of TCP semantics, higher latency at handover</td>
</tr>
<tr>
<td>Snooping TCP</td>
<td>“snoops” data and acknowledgements, local retransmission</td>
<td>transparent for end-to-end connection, MAC integration possible</td>
<td>problematic with encryption, bad isolation of wireless link</td>
</tr>
<tr>
<td>M-TCP</td>
<td>splits TCP connection, chokes sender via window size</td>
<td>Maintains end-to-end semantics, handles long term and frequent disconnections</td>
<td>Bad isolation of wireless link, processing overhead due to bandwidth management</td>
</tr>
<tr>
<td>Fast retransmit/fast recovery</td>
<td>avoids slow-start after roaming</td>
<td>simple and efficient</td>
<td>mixed layers, not transparent</td>
</tr>
<tr>
<td>Transmission/time-out freezing</td>
<td>freezes TCP state at disconnect, resumes after reconnection</td>
<td>independent of content or encryption, works for longer interrupts</td>
<td>changes in TCP required, MAC dependant</td>
</tr>
<tr>
<td>Selective retransmission</td>
<td>retransmit only lost data</td>
<td>very efficient</td>
<td>slightly more complex receiver software, more buffer needed</td>
</tr>
<tr>
<td>Transaction oriented TCP</td>
<td>combine connection setup/release and data transmission</td>
<td>Efficient for certain applications</td>
<td>changes in TCP required, not transparent</td>
</tr>
</tbody>
</table>

[Schiller]