Hierarchical Mobile IPv6
Implementation Experiences

Axel Neumann <neumann@ft.ee.tu-berlin.de>
Andreas Festag <festag@ee.tu-berlin.de>

TKN, TU-Berlin

Siemens IPv6 Workshop, June 3, 2002
Overview

- SeQoMo Project
- HMIPv6
  - Motivation, Objective, Signaling
- Implementation
  - Selection of Environment
  - Functionalities to be Implemented
  - Implementation Design
- Testbed Setup and Preliminary Performance Results
- Conclusion and Outlook
SeQoMo Project

• Objective:
  – Investigate suitability of IP-based networks to support of host mobility
  – Advanced mobility mechanisms, security and quality of service

• Selected features:
  – Optimize handover operation (hierarchical Mobile IPv6, multicast-based mobility), utilization of link-layer triggers
  – QoS-aware handover support (conditionalized handover)
  – Authorization to control access to different service qualities according to subscription context

Towards a **Secure QoS-enabled Mobility architecture**
HMIPv6: Motivation

- Assumptions for future all IP networks:
  - Many mobile nodes
  - Frequent handover due to smaller cell size
- Problems for MIPv6: Binding updates (BUs) are sent for every handover and traverse potentially the whole network
  ⇒ High signaling overhead
  ⇒ Long signaling latency
HMIPv6: Objective

- Assumption: Most movements are local
- Approach: Hierarchical concept
  - New entity: Mobile Anchor Point (MAP)
  - Logical tree structure in AN with MAP as root
  - Point of rerouting closer to MN
- Improvements: For local handover, signaling messages travel only up to MAP:
  ⇒ Less signaling latency
  ⇒ Less signaling overhead
HMIPv6: Signaling (1)

- CoA splits into Regional-CoA and Local-CoA
  - RCoA identifies region (AN)
  - LCOA identifies access point

- MAP discovery:
  - MN learns the availability of new MAP by MAP options piggybacked in router advertisements (1)

- Global movement:
  - MN registers LCoA with MAP (2)
  - MN registers RCoA with HA and CNs (3)
HMIPv6: Signaling (2)

- Local movement: Requires single registration
  - MN learns the availability of the same MAP (1)
  - MN registers new LCoA with MAP (2)
  - For HA and CNs the RCoA has not changed
Implementation: Environment

- No stable released HMIPv6 implementation

Mobile IP for Linux (MIPL):

- Project from HUT Telecommunications and Multimedia Lab.

- http://www.mipl.mediapoli.com/

- Pro:
  - Open, stable and well maintained IPv6 / MIPv6 implementation
  - Very responsive mailing list

- MIPL provides MIPv6 functionalities as a loadable kernel module which can be dynamically loaded, modified and reloaded into the running system
Implementation: Functionalities (1)

- MAP:
  - Advertise and propagate MAP options in router advertisements for MAP discovery
  - Same functionalities as HA:
    - Receive, process and confirm BUs
    - Maintain binding cache
    - Intercept packets destined to MNs RCoA
    - Tunnel intercepted packets to MNs LCoA

- Intermediate routers (IR):
  - Receive and propagate MAP-options in Router-advertisements
Implementation: Functionalities (2)

- MN:
  - Receive, process and maintain MAP options
  - Distinguish MIP and HMIP mode
  - If HMIP mode:
    - Distinguish local and global movement
    - Register at MAP with RCoA & LCoA
    - Register at HA and CNs with HoA & RCoA
Implementation: Selected Issues of Design

- Extensions and modifications of
  - MN data structures
  - MN event handling
  - MN policy handling
  - MN handover processing
Implementation: MN Data-Structure

List of MAPs

- MAP 1
  - RCoA
  - ADDR
  - CoA
  - lifetime
  - CONTEXT MAP
  - LCoA

- MAP 2
  - RCoA
  - ADDR
  - CoA
  - lifetime

Current MAP

- head_MAP

Next router (for handover only)

- next_MAP

Next MAP (for HMIPv6 handover only)

- head_AR

Current access router

- curr.Router

AR 1

- ADDR
- CoA
- lifetime
- CONTEXT MAP
- LCoA

Existing MIPv6

Extended for HMIPv6
Implementation: MN Event-Handling

- RA_rcv()
- NAck_rcv()
- mdetect()
- Ack_rcv()
- Timer()
- Router_Event()
- Router_update()
- Router_propose()
- change_Router()
- < if handover >
- set next_Router

Event handlers
Updates MN data structure
Selects next access router based on available information
Executes handover

Event handler existing in MIPv6
Event handlers modified for HMIPv6
Function calls modified for HMIPv6
Implementation: MN Policy-Handling

**Router_propose()**

- parse head_AR and set "Context-field":
  - is valid
  - well behaved
  - in home net
  - has MAP-Option
  - is current MAP
  - MAP confirmed
  - is current Router

Context-field

- Router with largest value of "Context-field" becomes handover target
- Ties are broken in preference of recently added routers
Implementation: MN Handover-Processing

Function calls modified for HMIPv6

- change_Router()
  - del_old_Routes()
  - set curr_Router
  - add_Router()

- MAP_reg()
  - set next_MAP
    - return
  - set curr_MAP

- HA-CN_reg()
  - return

Changes:
- Registers with the new MAP
- Deletes routes to old MAP
- Sets new access router
- Updates routing table
- Registers with HA and CNs
Testbed: Real Scenario

Testbed: System Model

HA

CN

(WAN)

MAPa

AR1

(AN)

AR2

MN

Ping pong handover

(Region 1)

(Region 2)

AR3

MAPb

WAN-Emulator

HA

CN

Datarate 10 / 100 Base T

WAN-manageable HUB

Region 1

Region 2

HomeNet

AR1

AR2

AR3

MN
Testbed: Tools and Setup

- **WAN-emulator**: SoftLink for IPv4
  - Tunnels IPv6 packets over IPv4
- **Traffic generator**: Ping6, SendIP6
- **Network monitor**: Tcpdump6
- **Movement emulation**:
  - Manageable SNMP hub
  - Use Netfilter to drop packets from certain AR
- **Emulate link-layer trigger**: LinkMonitor
  - Tool to monitor link connectivity
Testbed: Preliminary Performance Results (1)

- Experiment 1
  - Metric: Handover latency as duration of entire registration process (binding update to binding ack)
  - in case of global movement in HMIPv6 mode: duration from BU to MAP and BA from HA
  - RTT between MN and HA / CN: 90ms

<table>
<thead>
<tr>
<th></th>
<th>MIPv6</th>
<th>HMIPv6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Local Handover</td>
<td>91 ms</td>
<td>0.5 ms</td>
</tr>
<tr>
<td>Global Handover</td>
<td>91 ms</td>
<td>92 ms</td>
</tr>
</tbody>
</table>

Table 1: Handover latency
Testbed: Preliminary Performance Results (2)

- Experiment 2
  - Metric: Send ping6 packets from MN to CN and Count lost ping6 replies per handover
  - Constant generation interval of 10ms
  - RTT between MN and HA / CN: 90ms
  - LinkMonitor takes 20-30ms to detect movement

<table>
<thead>
<tr>
<th></th>
<th>MIPv6</th>
<th>HMIPv6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Local Handover</td>
<td>10</td>
<td>4</td>
</tr>
</tbody>
</table>

Table 2: Total packet loss per handover
Personal Implementation Experience

- Lack of protocol and implementation design for Linux IPv6/MIPv6 made the extensions to HMIPv6 a challenging task
- Complex structure of existing implementation
- Implementing in Linux kernel means:
  - Heavy fighting with the whole kernel, especially the:
    Network, IPv6 and MIPv6 stack
  - Hypertext cross reference tools to navigate the kernel was very helpful
  - Difficult debugging: the system tends to crash faster than any debugging message could reach the screen
- IPv6 tools for performance evaluation are hard to get
Hierarchical Mobile IPv6 conceptually improves handover latency and signaling overhead for basic Mobile IPv6

HMIPv6 functionalities based on MIPL have been implemented

Preliminary measurement results prove that HMIPv6:
- Outperforms MIPv6 in local movement
- Causes only small increase of signaling latency in global movement

HMIPv6 Implementation serves as basis for a prototype implementation of QoS-enabled handover scheme proposed in: draft-tkn-nsis-qosbinding-mipv6

Further measurements will be conducted