“An Analysis of the Flow-Based Fast Handover Method for Mobile IPv6 Network”

Jani Puttonen, Ari Viinikainen,
Miska Sulander and Timo Hämäläinen

Emails: janput@cc.jyu.fi, arjuvi@mit.jyu.fi,
sulander@cc.jyu.fi, timoh@cc.jyu.fi

Department of Mathematical
Information Technology
University of Jyväskylä
Finland

UNIVERSITY OF JYVÄSKYLÄ

Motivation 1/2

- Number of wireless terminals and applications increases
- Quality-of-Service sensitive applications and users
- Mobile IPv6 (MIPv6) standardized by the IETF to mobility management in IPv6 networks [1]
- Mobility made transparent to the application layer (L7)
- MIPv6 (L3) handover delay consists of
  - Movement detection (RSOL, RADS, NUD)
  - Care-of-Address configuration (Stateless or stateful address autoconfiguration, DAD)
  - Address registration (BU-process to HA and CNs)

Motivation 2/2

- MIPv6 is a major improvement for mobility (vs. MIPv4):
  - No need for foreign agents, route optimization, stateless address autoconfiguration, in-built IPSEC etc.
- But MIPv6 doesn't solve the handover delay in IPv6 networks!
  - Handover delay occurs when MN moves to a new subnet (MIPv6 procedures take too much time)
  - Delay might cause packet loss!
- Handover delay is not acceptable with real-time applications (VoIP, conferencing, etc.) and affects also TCP-based applications (data transfer)

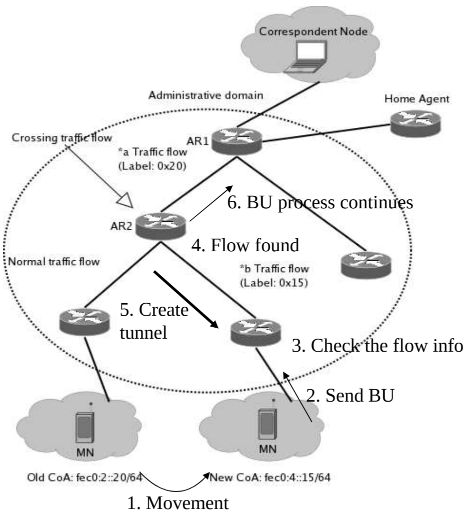
MIPv6 optimizations

- Several proposals for the handover delay minimization:
  - Hierarchical Mobile IPv6 [2]
  - Fast Handovers for MIPv6 [3]
  - Combination of the before mentioned
  - Solutions based on Multicast, L2 triggers, buffering, bi-casting, tunneling, etc.
- There is no perfect method. The methods improve some MIPv6 handover procedure (movement detection, CoA configuration, CoA registration)
- FFHMIPv6 [4] concentrates on CoA registration delay minimization (BU-process)

Method architecture 1/4

- Basic idea is to redirect the flows (heading for the old CoA of the MN) to the new CoA simultaneously with the BU process
- Enables the packet reception before the BU process finishes
- One binding update **can** update all the MN's connections simultaneously (correspondents and home agent)
- Uses native IPv6 features:
  - Flow information (source and destination addresses, flow label)
  - Extension headers: Hop-by-hop header
  - IPv6-in-IPv6 tunnels

Method architecture 2/4



Method architecture 3/4

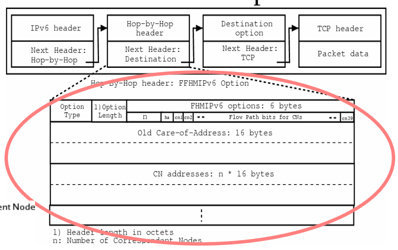
FFHMIPv6 processes:

1. MN moves to a new subnet and acquires a new CoA
2. MN sends a BU (including FFH info) towards the HA to register the CoA
3. In every router on the path to the HA, the router's flow information is checked (to find the flow heading for the old CoA)
4. The flow with the old CoA is found from router AR2
5. Tunnel created between AR2 and the new CoA. At this point forward the MN able to receive data!!!
6. BU-process continues (BU message towards the HA)
7. MN receives BACK from HA
8. MIPv6 handover finished! If we want to use route optimization, we still have to perform return routability and BU-processes to the CNs

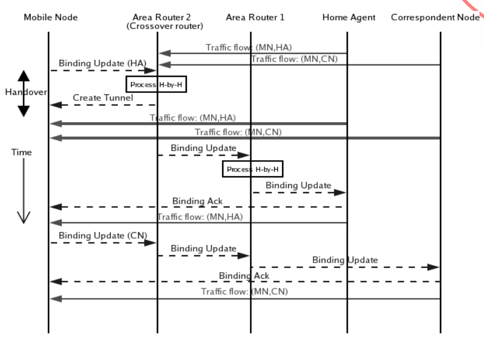
Method architecture 4/4

- Hop-by-Hop header added to BU message towards HA
- Every router processes the Hop-by-Hop headers

FFHMIPv6 option



Process diagram



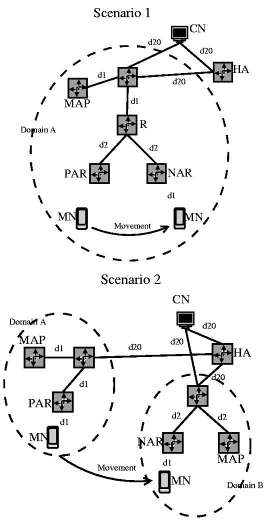
Benefits and requirements

- Key benefits (principles):
  - Simple architecture (always a good thing) and efficient
  - Relatively easy to implement (uses mainly native IPv6 features) and robust (no single point of failure (vs. HMIPv6))
- Uses existing network resources (router based functions: like tunneling and traffic engineering)
- Self-optimizing – network size is insignificant
  - Using the network topology hierarchical characteristic (all the traffic passing through one or more central router)
- Flow cache in every core router
- Tunneling capability in the core routers
- New Hop-by-Hop header identifier

Simulations

- Simulation environment:
  - Network Simulator 2 (ns-2.1b6)
  - Mobiwan Mobile IPv6 extensions
  - FFHMIPv6 implementation and some changes to the Mobiwan code
- Two simulation scenarios:
  - Worst and best scenario (topologically) for the FFHMIPv6 method [5]
  - The effect of the distance of the HA and CNs in a “typical” hierarchical network environment [6]

Simulation scenario 1

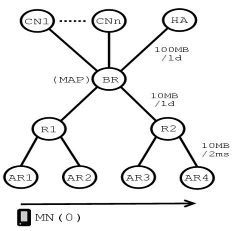


- Scenarios simulated by using MIPv6 and the FFHMIPv6 method
- Scenario 1 (best case):
  - MN moves within a domain having already established traffic flow
- Scenario 2 (worst case):
  - MN moves to a domain that has no previous traffic flow

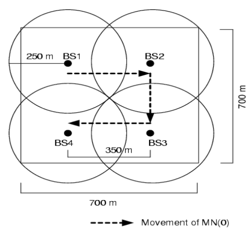
RESULTS OF THE NS2 SIMULATIONS

	Scenario 1		Scenario 2	
	MIPv6	FFHMIPv6	MIPv6	FFHMIPv6
Handover time (ms)	58.8	13.8	65.2	65.2
Packet loss (pkts)	5	1	6	6

Simulation scenario 2 1/2



Simulation scenario

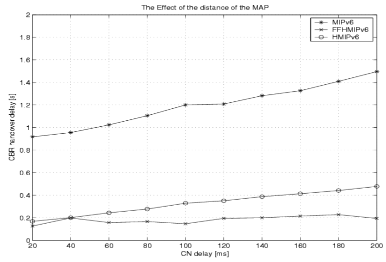


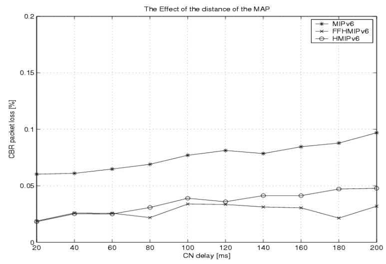
Base station location

- Simulation set
  - BSs connected to different ARs (BS1-AR4). BR acting as a MAP for HMIPv6, it is also a crossover router for traffic
  - UDP based traffic (CBR 80Kbps, packet size 1000B, interval 0.1s)
  - Delay between R1,R2 and BR varied (20-200ms) in 20ms increments
  - MN moves through all BS areas

Simulation scenario 2 2/2

- MIPv6 performance depends on the distance of the HA and CNs
- HMIPv6 performance depends on MAP location and network size
- FFHMIPv6 provides constant handover delay, but is somewhat related to network topology
  - MIPv6: delay < 1.5s, packet loss < 0.1%
  - HMIPv6: delay <0.4s, packet loss <0.05%;
  - FFHMIPv6: delay <0.2s, packet loss <0.03%





Conclusions

- FFHMIPv6 method will decrease the handover delay in IPv6 networks (network size not a critical factor, self-optimizing behaviour)
- Likely that the performance is near the best case scenario in large networks (established traffic flow within few hops)
- Robust (no single point of failure)
- Implementation requires only few changes to current MIPv6 implementation (using native and existing features). Uses limited resources on the implementing devices.

On-going and future work

- Implementation of FHMIPv6 to ns-2
- Larger simulations to analyze the methods
- Security in FFHMIPv6
- Implementation of FFHMIPv6 to MIPL (Mobile IPv6 for Linux)
- Intelligent vertical handover system

References

[1] D. Johnson, C. Perkins and J. Arkko, "Mobility Support in IPv6", IETF RFC 3775, June 2004.

[2] H. Soliman, C. Castelluccia, K. El-Malki, L. Bellier, "Hierarchical Mobile IPv6 Mobility Management (HMIPv6)", IETF draft, work in progress, 2004.

[3] R. Koodli, "Fast Handovers for Mobile IPv6", IETF draft, work in progress, 2004.

[4] M. Sulander, T. Hämäläinen, A. Viinikainen, J. Puttonen, "Flow-based Fast Handover Method for Mobile IPv6 Network", in Proceedings of the 59th Semiannual IEEE Vehicular Technology Conference (VTC'S04).

[5] J. Puttonen, A. Viinikainen, M. Sulander, T. Hämäläinen, "An Analysis of the Flow-based Fast Handover Method for Mobile IPv6 Network", in Proceedings of the International Conference on E-Business and Telecommunication Networks (ICETE2004).

[6] J. Puttonen, A. Viinikainen, M. Sulander, T. Hämäläinen, "Performance Evaluation of the Flow-based Fast Handover Method for Mobile IPv6 Network", **to be published** in IEEE Vehicular Technology Conference (VTC'F04).