A client-driven mobility frame system – Mobility management from a new point of view

Benedek Kovács, Péter Fülöp

Budapest University of Technology and Economics, Department of Telecommunications
{bence, fepti}@mcl.hu

Keywords: mobility management, CMFS, modelling, client-based, handover

In this paper a new mobility management approach is introduced. The main idea in this approach, that not the network but the mobile node should manage the mobility for itself (similarly to the IP concept). The network nodes provide just basic services for mobile entities: connectivity and administration. We construct a protocol called the Client-based Mobility Frame System (CMFS) for this mobility environment. We propose some basic mobility management solutions that should be implemented in the mobile clients and provide details about a working simulation of a complete Mobility Management System. Examples of mobility management approaches such as the centralized- and hierarchical or cellular-like ones are also defined and hints are given what kind of algorithms might be implemented upon the Client-based Mobility Frame System. After the theoretical analysis a simulation shows the applicability of the newly introduced protocol framework.

1. Introduction

Seamless information mobility is a requirement in the today's world. Although there are many operating solutions there is still need for IP mobility since IP is the most widespreadly used protocol. The communicating equipments are identified with their permanent IP address and the communication is done on IP networks. Many works have discussed the problem of managing the movement of the clients since the Internet was designed to be static and does not support mobility by itself. There are different solution proposals for the problem and all of them have their drawbacks and good features.

If one takes a close look at these systems they always deal with the tradeoff between complexity (simplicity) and optimality. Naturally, this can not be resolved but we will transform it into another dimension: from network level to individual level.

In this paper we introduce an agent based mobility management strategy. This is an alternative point of view and it could be easier to implement our solution, than the classical ones. We do not say that we have found the optimal system to provide IP or other kind of mobility but we will come up with a new idea and framework which is very different from the classical approaches and can be the most cost-efficient in many cases.

The basic idea is that, unlike in the GSM or Mobile IP (both IPv4 and IPv6), the network will no longer have to provide any logic for the management algorithm. The whole network can remain simple and the nodes will only have to handle simple commands by recognizing, executing and forwarding simple messages generated by the mobil entity itself. The management system is implemented in the mobile client, consequently each node is able to choose the most suitable mobility for itself on the same network.

We show how to apply the classical strategies like cellular or hierarchical approaches to our system. First we will present a protocol description and define the Client-based Mobility Frame System then we give simple mobility applications like the Mobility Management Systems itself.

2. Client-driven Mobility Frame System

We have introduced the new idea and explained the basics of its operation. In this Section we will define a Client-driven Mobility Frame System (CMFS) specifying the basic roles in the network and what capabilities the fixed network nodes are required to have to be able to communicate with the moving entity. A simple method will be given for the mobile node to discover the service network and build up its own logical network. We will handle the cleansing of the service network database.

2.1. Some basic notations

Since there are many kind of notations in this field, to avoid misunderstandings, some of the basic ones we use are defined here. The mobility model will be the same abstract one as in one of our previous works [1]:

- The *Mobile Nodes* (MN, alias mobiles, moving entities) are the mobile devices who want to communicate to any other mobile or fixed partner.
- There are *Mobility Access Points* (MAP) as the only entities who are capable to communicate with the Mobile Equipments. (Note: mobility does not necessary imply radio communication. It means only that the Mobile Node changes its Mobility Access Points and when it is attached to one, communication between them can be established).
- The *Mobility Agents* (MA) are network entities running the mobility management application.

- There is a *Core Network* that provides communication between the Mobility Access Points and has a structure that can be described with a graph. Vertices are either Mobility Access Points or Mobility Agents other serving nodes who is not part of the mobility management application and the edges can be any kind of links (even radio links) for the data communication between the vertices.
- *Home Agent*, a special MA that is a kind of basis to the Mobile just like in the Mobile IPv4 [6] or the Mobile IPv6 [7] case or the HLR in the GSM case. Whenever the MN is paged its exact or approximate location can always be found in the database of this node.

2.2. The idea

The client driven mobility management we introduce is inspired and based (but does not depend!) upon the fact that a mobil user typically moves within a range of access points and rarely leaves to far away agents. In order that the mobile could manage its own mobility it has to maintain a database of the nodes it communicates with. This is called the Logical Network (LN). The MN always should be able to have an up-to-date information of the nodes of this network. The size of this depends on the algorithm the mobil uses.

To give an example we will show later that to implement a basic Mobile IPv4-like solution on our framework system the MN only has to maintain information about 3 (or even only 2) nodes in the network so for a node with very limited capacity this can be a good enough solution.

We want to point out that the most important advantage of our solution is that the service providers do not have to choose an exact mobility approach, which could be very inefficient. The mobile nodes in CMFS can choose the optimal algorithm for themselves, thus the mobility solution can be the most cost-efficient and adaptable for various circumstances.

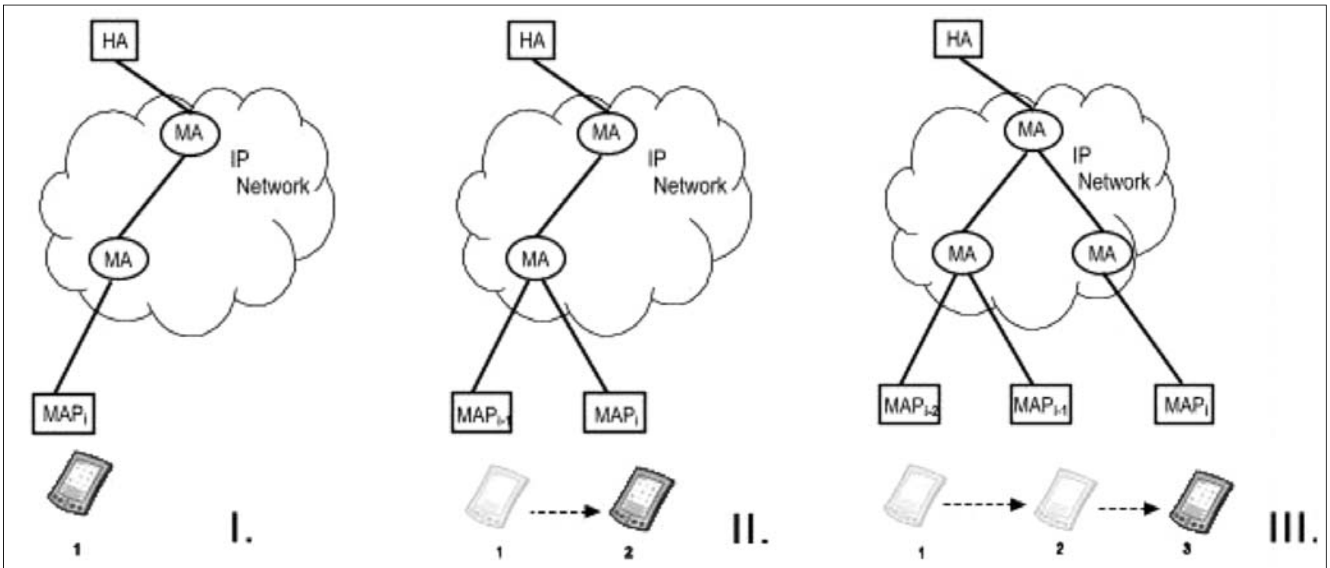
2.3. Network discovery

In order to implement more complex mobility managements, the MN should construct and maintain a larger logical network, it have to get to know the network entities. There are several algorithms to discover heterogeneous IP networks. In this paper, we do not focus on the selection of the most efficient or optimal one. We introduce a simplest method and aim to prove that the system we propose actually works.

Although the simplest would be to use the special IP packet options value for network layer packet tracing, it is not feasible since most network entities are not able to interpret these packets due to lack of implementation and poor specification. The other possibility is to use the *traceroute* application or anything like that which does not depend on any facilities and use the TTL (Time To Live) function of the IP. By using small TTL values which quickly expire, traceroute causes that the routers along a packet's normal delivery path generate automatically an ICMP Time Exceeded message. We use a similar method in our own protocol specification. We implement it in the update procedure. Let us examine this update and network discovery function without a precise protocol specification. (As for the protocol implementation, see the next section).

For a basic algorithm like MIP [6] at the first step, the Mobile Node (MN) registers with its Home Agent (HA) that is the first Mobility Access Point (MAP) in the Logical Network (LN). Then the MN moves to a Foreign Network (FN). It gets a Care-of-Address (CoA) as in the MIP solution but also tells the MAP what it should do: whether it should register with the Home Agent (HA) or with any other Mobility Agent (MA) etc. If the MN wants to know the path to the HA, then it sets a bit in the *register* message, which triggers *reply* messages with a timestamp from all, or only from specific MAs on the way. The timstamp could be used to determine the weight of the links towards the MAs. The MN records the discovered MAs to its database and to the Logical Network it maintains.

Fig. 1. The logical network build-up



Based on this information it can implement more complex management algorithms. When it moves to another network and communicates with another MAP it can delete the former one from its database and proceed. This is how the MN can maintain the whole network it has to know. More complex Network Discovery procedures are going to be discussed when needed.

2.4. The requirements for the network MAs

To be able to serve the MNs and their algorithms we define requirements for the network. Once, the MAs are aware of all of them, an MN can use any kind of Mobility Management Strategy (MMS) for itself. This also means that different terminals are allowed to use the most suitable MMS for themselves.

Since all the management is MN initiated, the MA has to provide a kind of routing function. The HA should always know where to route a packet towards the MN, or drop the call. There should be a database registry for this, for example an association between the MNs permanent IP (Care of Address, CoA) and routes: where to forward the packet towards the CoA. All the MAs should work in a similar way. Once the MN with its CoA is paged at an MA it should route the packets to the MAP of the MN. If there is no route to the MN it should simply drop the packets. How can an MN register to an MA? When it attaches to a new MAP after a successful handover it naturally registers there. It also adds the information whether this MAP should continue the registration process to an MA in an upper level or not.

Let us construct such a message:

```
[Dst: MAPi, Src: MN, Actions: Register MN to MAPi via MN;
  [Dst: MAj , Src: MAPi, Actions: Register MN to MAj via MAPi,MAPii,MAPiii;
        [Dst,Src,Actions: ; ;
              [ ...
              [Dst: HA, Src: MAn, Actions: Register MN to HA via MAn]
              ]
        ]
  ]
].
```

What the MA should do is to understand this message and maintain the following entry in its database: if the paged node is MN then it should be searched via HA, MA_n, ..., when MN is searched at MA_j then MA_j knows that it can be reached MAP_i, MAP_{ii}, MAP_{iii} meaning that the packet is routed to all the 3 nodes representing a CIP-like algorithm [2]. If there is no such multiple route and the messages do not always contain the HA, then a HMIP-like protocol [3] is implemented. If there is an update that goes from MN directly to HA, then a MIP-like approach [6] is implemented, if these last two kind of messages are mixed then a DHMIP-like approach [5] is presented. If the node sends messages like

```
[Dst: MAPi, Src: MN, Actions: Register MN to MAPi via MN;
  [Dst: MAPi¡1 //The former node//,
        Src: MN, Actions: Register MN to MAPi¡1 via MAPi
  ]
]
```

then a HAWAII-like protocol [8] is implemented.

In case of wireless tracing (for example LTRACK [4]), two different messages would be sent:

```
[Dst: MAPi, Src: MN, Actions: Register MN to MAPi via MN;]

[Dst: MAPi¡1 //The former node//,
Src: MN, Actions: Register MN to MAPi¡1 via MAPi]
```

We have provided an implementation example that can be modified after reasonable discussions but just like IP or any kind of protocol it should be standard in any network the MN wants to communicate in.

2.5. CMFS Protocol

For specification of the aforementioned command structure we developed an application layer protocol called CMFS Protocol (CMFSP). A CMFSP message is carried in UDP packets. We have chosen it instead of the TCP because the TCP does not operate well with the radio interfaces. The TCP conceives the high bit error rate of the radio channel as congestion, and decreases the window size that ends in significant speed fall-off. For this reason the mobility applications generally use UDP to the communication.

The CMFSP message structure follows strict rules as it can be seen in *Fig. 2*.

The header contains 4 fields of 1 byte elements, a *type, length, flags* and *number of actions* and a 4 byte element. Presently two different types of CMFSP messages are differentiated, a *request* and *reply* message. The *length* shows the full length of the CMFSP packet included the header. The *destination* tells the node that it has to process the message. The first bit of the flags field is the trace bit (F). If it is set to 1, it means the first MA on the way to the HA must send a CMFSP reply message, in order that the MN be able to build its Logical Network. The second bit of the flags is the specified trace bit (S). In this case only the MAs must send reply, which are labelled in the CMFSP message. The third bit (L) means all the MAs on the way to the HA must send a CMFSP reply message. The last bit of the flags (C) is set if MN wants to get capacity information from the MAs.

The *payload* of the CMFSP message contains the actions, which have to be accomplished at the specific nodes. One can see that there are three different kinds of actions defined. The first is the *Register* action that indicates a route registration in the given MA via the given destination to a specified target. The second type is the *Delete* action that erases the specified registered data from the MA database. The *Send* action type instructs the MA, that the payload of the action field has to be send as a CMFSP message.

3. Examples of mobility management strategies implemented in CMFS

The good thing in the Client-based Mobility Frame System is that not just all the MNs can use different handover management strategies but a single MN can switch between them easily upon request or in a seamless way if implemented so. What the MN does is collecting the network parameters and makes decisions upon them and commands the network nodes accordingly with the messages defined above. Here we will show how to implement the most common mobility approaches like solutions into our Client-based Mobility Frame System.

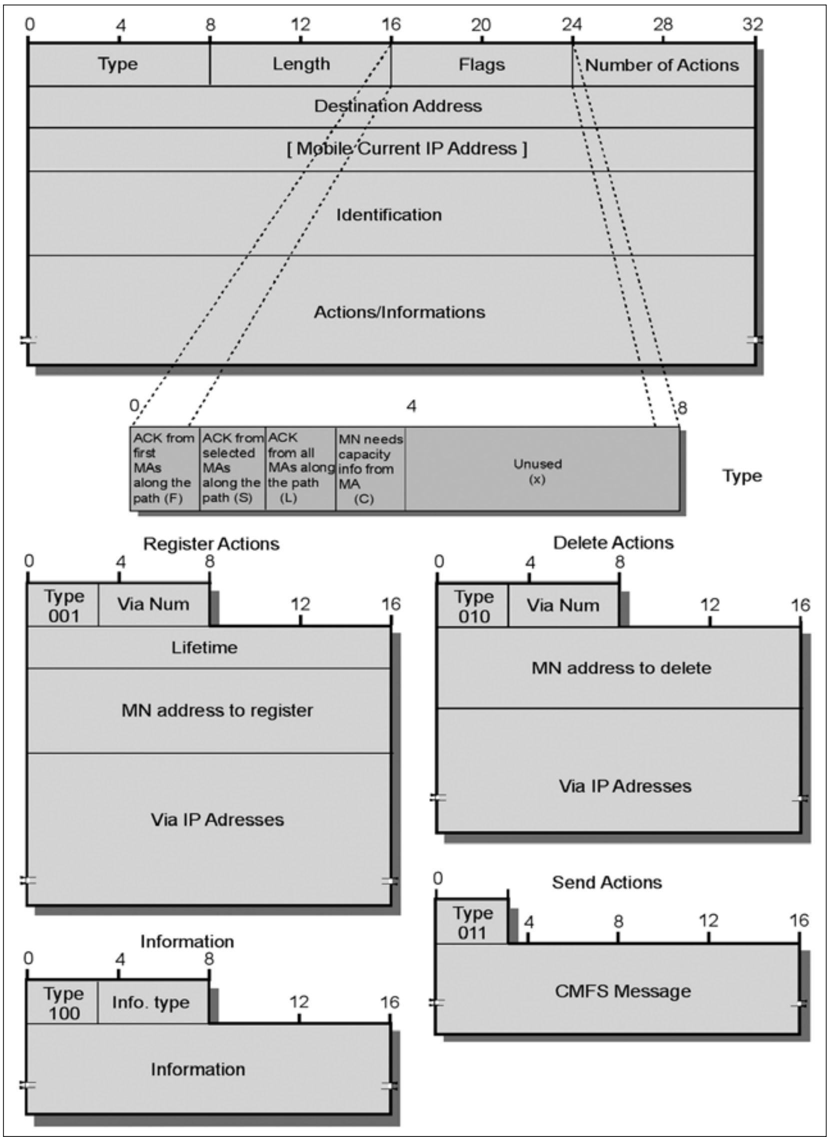
3.1. Personal Mobile IP – PMIP

The operation of Personal Mobil IP is simple and easy. Once the MN attaches to MAP it registers itself to the HA. The operation is very similar to MIP and has a great advantage. The MN has to make no extra computation and has to maintain no extra database while there are always a few routes in the MAP.

[Dst: MAPi, Src: MN, Actions: Register MN to MAPi via MN;
[Dst: HA, Src: MAPi, Actions: Register MN to HA via MAPi];
[Dst: MAPi¡1, Src: MAPi, Actions: Delete MN in MAPi¡1 via MN]
].

Where the second message is needed only if clearing the network is up to the MN unlike in MIPv4. This solution is referred as pure PMIP (P-PMIP).

Fig. 2. CMFSP message structure



The simple PMIP protocol operates alike MIP and has approximately the same capacity consumption as well as we will see later. We would like to point out that the MN has to maintain a Logical Network of three nodes only. However, a great benefit of our proposal is that any MN can implement different version (e.g. soft handover) of the protocol without any modification in the network entities.

Then the Extended PMIP (E-PMIP) is an example of extension of PMIP when there is no packet loss and no obsolete routes in the databases of the MAs but of course the messages are more complex.

One can see what happens in case of a handover on *Fig. 3*.

[Dst: MAPi, Src: MN, Actions: Register MN to MAPi via MN;
[Dst: MAPi¡1, Src: MAPi, Actions: Register MN in MAPi¡1 via MAPi;
Delete MN in MAPi¡1 via MN;
[Dst: HA, Src: MAPi¡1, Actions: Register MN to HA via MAPi;
Delete MN in HA via MAPi¡1;
[Dst: MAPi¡1, Src: HA, Actions: Delete MN in MAPi¡1 via MAPi
]]]]

The performance analysis can be found in Section 6.

3.2. Personal Hierarchical Mobile IP – PHMIP

The operation of a HMIP micro-mobility (talking about an only two-layered hierarchy) would pose the question: which node should be the MA in the hierarchical mobility approach. We suppose that seeing the traceroute messages, the MN can decide it. The messages are again simple and easy to construct.

More problems arise when talking about multiple layered hierarchical solutions. The MN has to make complex calculations for setting up the network tree but still the only problem will be to locate the logical junctions in the node (those MAs which are not MAPs). However, once this is solved the implementation again easy since there is no need to configure the network itself and implement the protocol in a static way.

Now let us give a simple method to choose the MAs that will be used to construct the hierarchy tree of the network. At the beginning the MN is attached to its HA then it moves to another MAP. The MN records all the MAs along the way (from the MAP to HA). Then when it makes a handover it records the way again. The first common element of the route (from the MN) is then dedicated to be a Hierarchy Point.

This method is very easy to implement and rather simple. We show in our simulation work that it still outperforms the basic protocols.

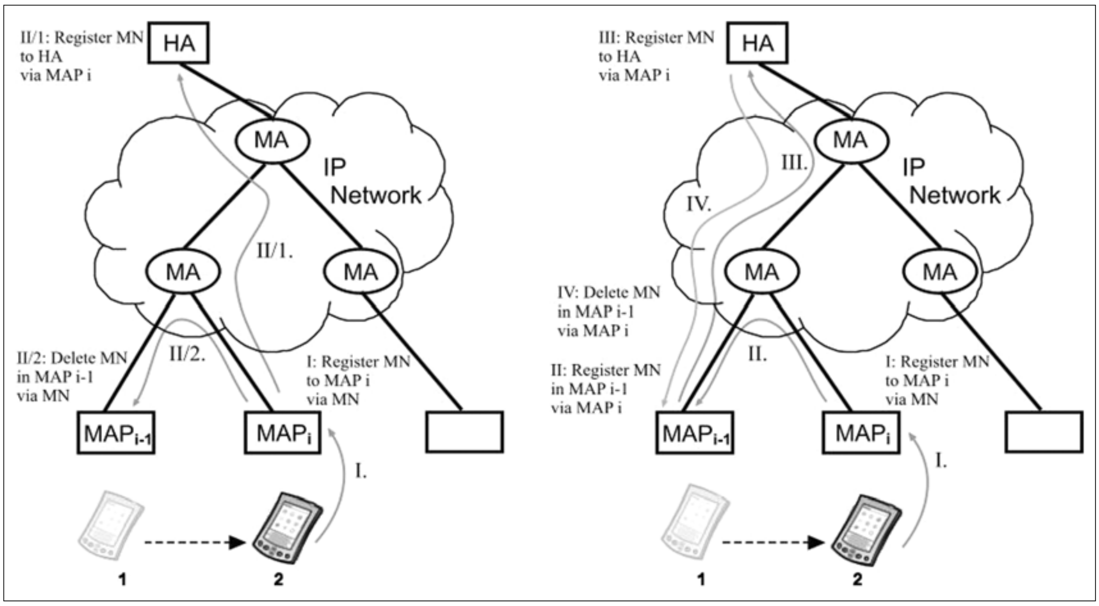
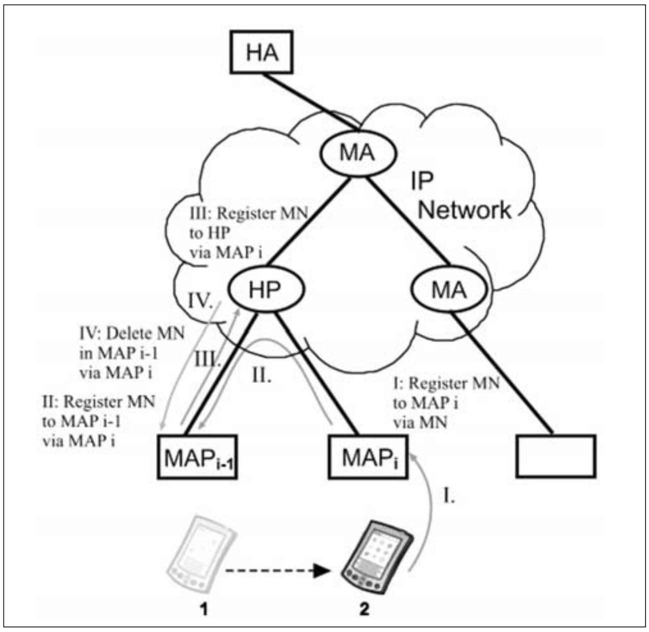


Fig. 3. In the left figure one can see the basic P-PMIP protocol, while the figure on the right depicts the operation of the action-linearized Personal Mobile IP Mobility Management System (E-PMIP) with soft handover mechanism.

Fig. 4. The operation of PHMIP



3.3. Personal Tracking Mobile IP – PTMIP

A tracking-like (see *Fig. 5*) solution would be again easy to implement. In this case the tracking handover is introduced when the MN orders the new MAP to report always only to the previous MAP it was attached to like in the DHMIP [5] or LTRACK [4] protocols.

When the MN is paged the message is sent through all the nodes along the way. For this reason, after a number of tracking handovers the MN performs a normal handover i.e. registers back with the HA (or to some hierarchy point in a more complex solution).There are many proposed methods to decide between the two types of handovers. In our simulation we implemented a simple suboptimal solution when the MN registers back at every *i*th step.

3.4. Personal Cellular Mobile IP – PCMIP

Since the widespread use in GSM the cellular solutions became popular in most mobility applications. The idea is to avoid registrations when the MN moves within a given set of MAPs but then it has to search for it at each MAP when it is paged. There is a extensive literature of cell forming algorithms. We give an alternative one.

We want to point out that in this case the paging areas are different for each MN and are formed in an almost optimal way by each MN individually. We expect better performance in large networks. The MN should send registration messages only when it moves to a new Paging Range (PR).

In this case it orders the leader of the new Paging Range to register at an upper level that the MN is in the PR. The MN also tells the IDs of the MAPs in the Paging Range (PR) to the leader of the PR so that the latter be aware who to broadcast the messages when the mobile is paged.

The following message tells to the specific MAP (the leader) the MAPs (MAP_i, MAP_{ii}) belonging to that given PR:

[Dst: MAPleader //The leader of the paging area//,
    Src: MN, Actions: Register MN to MAPleader via MAPi, MAPii, ... ,
        [Dst: HA, Src: MAPleader, Actions: Register MN to HA via MAPleader
        ]
]

The problem to solve for cellular algorithms is the problem of forming he Paging Ranges. Forming the cells at an optimal cost using the total frequency of handovers on aggregate level (not individually for each MN) is NP hard. Consequently, the problem is NP hard for only one MN too. However, there are alternative solutions giving a solution that is good enough.

4. Simulation and numerical results

We have made a simulation to show at first that or proposed method actually works and secondly to compare it with existing technologies. The simulation was written in the open source OMNet++ [13] using C++ language.

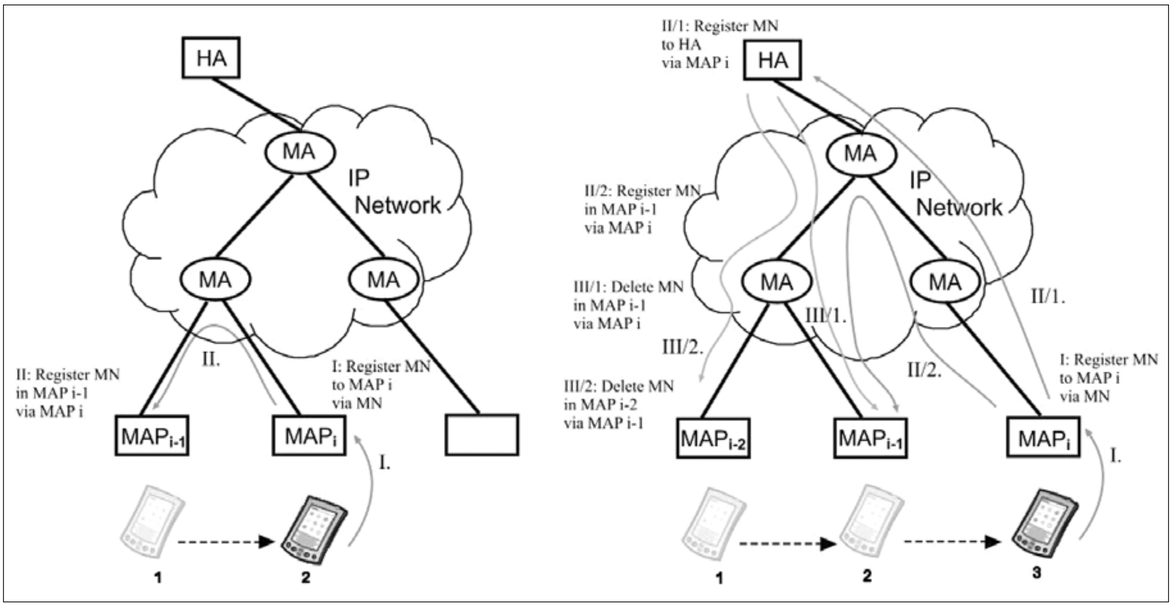


Fig. 5. The operation of the Personal Tracking Mobile IP protocol. The tracking handover is depicted in the figure on the left while the figure on the right is about the normal handover.

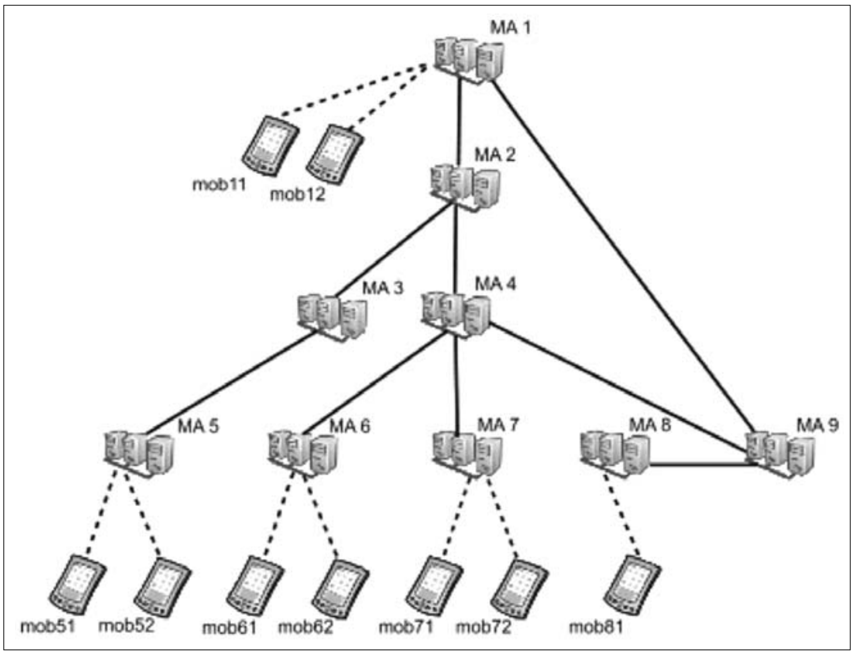


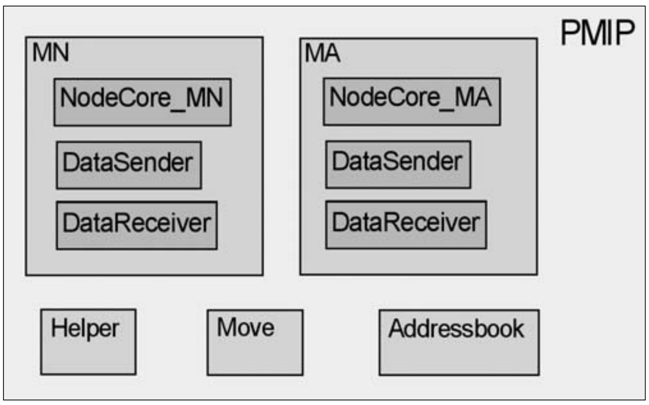
Fig. 7. The test-network used in the simulation

The simulation consists of two main modules, namely MN and MA, and some other simple components that are needed to model the operation environment *(Fig. 6)*. The two main modules have similar internal structure. Both has a *DataSender* and a *Data Receiver* to be able to send and receive messages while their logic is hidden in *NodeCore MN* and *NodeCore MA*, respectively.

The whole CMFS protocol is implemented in the Node Core components. The NodeCore MN constructs CMFS messages, maintains a database and builds up the Logical Network. The NodeCore MA understands the CMFS messages and executes the actions, maintains the database and routes the messages and packets using it.

The DataSender module creates traffic in the network to a random target and at random times while the DataReceiver is responsible for receiving and analyzing it. The number and size of packets, the frequency of sending data and the possible targets for a node can be set as a parameter of the simulation. The receiving side measures the average number of handovers, number of arrived/sent/lost packets and their averages in 1 min but can be extended to record other QoS parameters like delay or jitter too.

Fig. 6. The component structure of the simulation of CMFS written in OMNet++



The *Addressbook* module is the template for the databases in the MAs. The module Move is responsible for the directions and frequency of movement of the MNs. The Helper component implements some functions and objects that are not logically part of any of the above ones.

We have constructed a virtual test environment consisting of 9 MAs and 9 MNs with the initial MN distribution depicted in *Fig. 7*.

We have run the simulation on various mobility parameters for all the algorithms separately. All the nodes made calls according to a Poisson process to random targets with a biased uniform distribution so about 80% of the calls were terminated at mobile clients. The mobility ratio (number of handovers per received call) was varied to show how it affects the performance.

The performance of the protocols is depicted in *Fig. 8*. However at low mobility level (when there are only a few handovers between two calls) E-PMIP is better than the classical MIPv4 but as the mobility ratio increases the protocol performs worse in terms of signalling load on the network. It is because it requires more operations and messages in the network to provide better QoS parameters. We can see that the P-PMIP is always better than the MIPv4. This is because if we look at the two protocols both have the same signalling strategy but MIPv4 needs Agent Advertisement messages to maintain connectivity while in the client based system it can rely on lower layers. We can conclude that the basic solutions work at approximately the same costs. However, E-PMIP shows that it is possible to improve the performance while not changing the protocol at all (only on the MN side).

In the simulation we implemented the PTMIP also, and we examined it with different tracking handover numbers. *Fig. 9* shows the results. More interesting simulations can be applied in the OMNET++ framework developed by us, but the most important conclusion can be seen: the CMFS is correctly works, and all the well-known mobility protocols can be implemented in it.

5. Conclusion

We have introduced a mobility management system that solves IP mobility from a very different point of view than any other mechanism known so far.

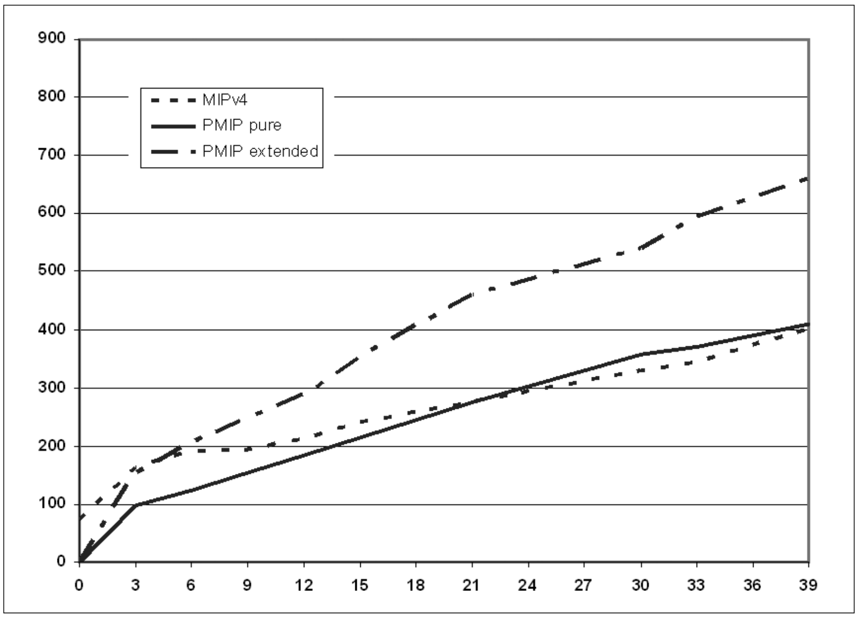


Fig. 8.
Comparison of the three mobility management systems MIPv4, PMIP pure and PMIP extended.
The horizontal axis shows the number of handovers between two arrived calls while the number of bytes transmitted on the network by each protocol is presented on the vertical axis.

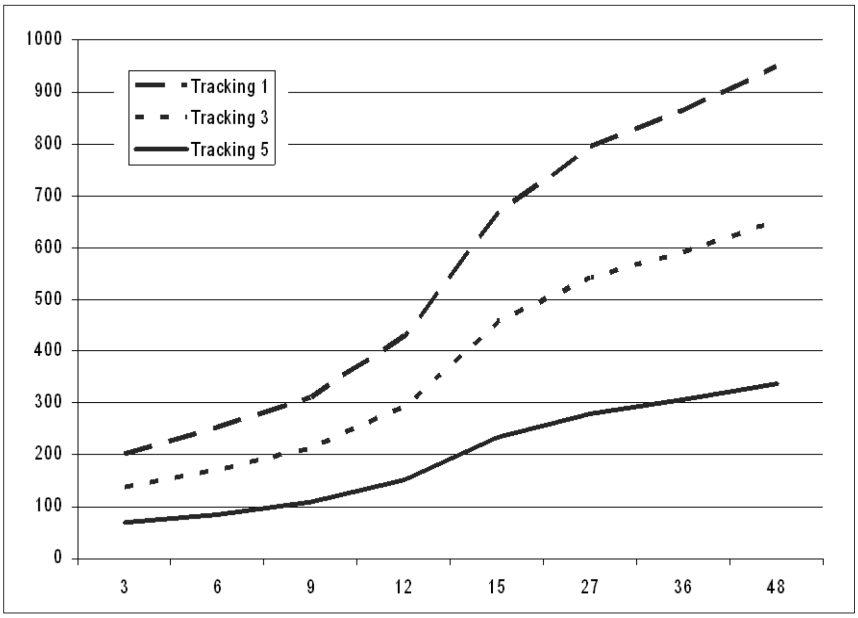


Fig. 9.
This figure depicts the performance of three tracking-like approaches namely PTMIP with 1, 3, 5 tracking handovers. Note that for this simulation a couple of additional links were inserted into the network.

We have shown example algorithms taking ideas from classical solutions. We prepared a simulation and tested our protocol in operation. Using it we compared the performance of some basic solutions and we have shown that extensions may be beneficial for both the MN and the network. Further extensions are possible: since the MN records the details of a MAP it can also perform quality measurement or reliability measurement, thus classify the MAPs and networks and use this information in the future (for example when multiple MAPs are available).

We have shown how CMFS would work over IP. However, it is rather simple to extend the whole to IMS too.

Authors

BENEDEK KOVÁCS received his M.Sc. degree in Technical Informatics from the Budapest University of Technology and Economics in 2006. Currently he is a PhD student on the faculty of Mathematics at the same university on the Department of Mathematical Analysis and member if the High Speed Network Laboratory, Hungary. Currently he is doing research work at Ericsson Telecommunications Hungary, Traffic Lab. His research interest is IP Mobility, Overload and Load Regulation methods and intensity estimation of Stochastic processes in telecommunications and parameter estimation of dynamic processes. The topic of the upcoming PhD theses is the optimal control of dynamical stochastic systems.

PÉTER FÜLÖP received his M.Sc. degree in Technical Informatics from the Budapest University of Technology and Economics in 2005. Currently he is a PhD student on the faculty of Informatics at the same university on the Department of Telecommunications. He is working as a system test engineer at Ericsson Telecommunications Hungary at Research and Development Department. His research interest is IP mobility, interworking of heterogeneous mobile networks and movement modeling in cellular, mobile networks. The PhD thesis is going to be written on Complex Mobility Management Systems and Applications.

References

[1] Kovács B., Fülöp P., Imre S.
"Extended Mobility Management Framework",
International Conference: 6th Computer Information Systems and Industrial Management Applications Conference Proceedings, CISIM 2007,
pp.191–196., 2007.

[2] A. T. Campbell, J. Gomez, A. G. Valkó,
"An Overview of Cellural IP",
Wireless Communications and Networking Conference IEEE, Vol. 2, pp.606–610., 1999.

[3] C. Castelluccia,
"A Hierarchical Mobile Ipv6 Proposal",
INRIA Technical Report, pp.48–59, 2000.

[4] Kovács B., Szalay M., Imre S.,
"Modelling and Quantitative Analysis of LTRACK – A Novel Mobility Management Algorithm",
Mobile Information Systems, Vol. 2, No.1,
pp.21–50., 2006.

[5] W. Ma, Y. Fang,
"Dynamic Hierarchical Mobility Management Strategy for Mobile IP Networks",
IEEE Journal of Selected Areas In Communications, Vol. 22, No. 4,. pp.664–676., 2004.

[6] C.E. Perkins,
"Mobile IP",
IEEE Communications Magazine, Vol. 40, No. 5,
pp.66–82., 2002.

[7] W. Fritsche, F. Heissenhuber,
"Mobile IPv6 – Mobility Support for the Next Generation Internet",
IAB GmbH, 2000.

[8] R. Ramjee, T.La Porta, S. Thuel,
K. Varadhan, L. Salgarelli,
"A Hierarchical Mobile IP Proposal",
INRIA Technical Report, 1998.

[9] S. Das, A. Misra, P. Agraval, S. K. Das.
"TeleMIP: Telecommunications-Enhanced Mobile IP Architecture for Fast Intradomain Mobility",
IEEE Personal Communications, Vol. 7, No. 4.,
pp.50–58., 2000.

[10] Szalay M., Imre S.,
"Hierarchical Paging – A novel location management algorithm",
ICLAN'2006 – International Conference on Late Advances in Networks, 6-8 Dec. 2006, Paris, France.

[11] K. D. Wong, W. W. Lee,
"Likelihood of Lost Binding Updates when Mobile Nodes Move Simultaneously",
MOMM'2005 – Third International Conference on Advances in Mobile Multimedia, 2005, pp.9–19.

[12] Kovács B., Fülöp P., Imre S.,
"Mobility Management Algorithms for the Client-driven Mobility Frame System – Mobility from a Brand New Point of View",
MOMM'2008.

[13] http://www.omnetpp.org/