Mobility Management Algorithms for the Client-driven Mobility Frame System - Mobility from a Brand New Point of View

Péter Fülöp
Budapest University of Technology and Economics, Department of Telecommunications
H-1521 Pf. 91.
Budapest, Hungary
fulopp@hit.bme.hu

Benedek Kovács
Budapest University of Technology and Economics, Department of Mathematical Analysis
H-1521 Pf. 91.
Budapest, Hungary
bence@mcl.hu

Sándor Imre
Budapest University of Technology and Economics, Department of Telecommunications
H-1521 Pf. 91.
Budapest, Hungary
imre@mcl.hu

ABSTRACT

In this paper we introduce some example algorithms that can work with the Client-based Mobility Frame System presented recently by us. In this solution all the logic and structure is transferred from the network to the Mobile Node and thus each individuum can use the resources in a personalized and efficient way. In the present work modeling and detailed discussion on the parameters of the algorithms is given and comparison to existing mobility approaches and protocols is done. We prepared a simulation to test our protocol and to back up the proposals we provide the reader with simulation results. We stress that one the most important benefit of our findings is that all the MNs can run different management strategies and can optimize mobility for themselves.

1. INTRODUCTION

Seamless information mobility is a requirement in today's world. Although there are many other alternative operating solutions there is still a need for IP mobility since IP is the most spread protocol. The communicating equipments are identified with their permanent IP address and the communication is done on IP networks. Many works has 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 mobile node level.

In this paper we present our agent based mobility management solution and propose a couple of strategies one can implement to the mobile equipment and then have very efficient mobility solution. Our alternative point of view is easy to implement and does not require complex network setups to operate. We do not say that we have found the optimal system to provide IP or other kind of mobility but we come up with a brand 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 classical Mobile IP solutions, the network will no longer have to provide any logic for the management algorithm. The whole 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 accordingly 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 the simple MIP, hierarchical, tracking or cellular approaches to our system and furthermore we propose an algorithm that creates cell structures efficiently and individually for each MN.

The structure of the paper is the following. First we will present a protocol description i.e. the definition of the Client-based Mobility Frame System and than we give simple mobility applications what is the Mobility Management Systems itself. We give example algorithms to implement based on ideas from classical mobile IP solutions. A proposed nearly optimal solution for some parameters of given algorithms are discussed in detail and analyzed mathematically. Then we present the details of our simulation and we conclude the whole with some numerical results, figures comparing the new solution to existing ones.

2. NETWORK REQUIREMENTS

Here we give a short description of our solution named the Client-based Mobility Frame System (CMFS). This system provides an interface to the Mobile Nodes (MN) what they can use to manage their own mobility with any kind of algorithm implemented. In this Section we summarize the requirements for the network nodes. Note that not all the nodes in the network must necessarily have the CMFS implemented on it but of course the MN will recognize only them. These nodes called Mobility Agents (MA) or Mobility

MOMM2008 November 24-26, 2008, Linz, Austria

Access Points (MAP) where the latter is the access point to the network for the MN. (Mostly we can think of a MAP as a Foreign Agent in the MIPv4 [5] terminology.)

2.1 Non-mobility related requirements

At first some basic assumptions are made as basic requirements.

The most important is that all the nodes in the network should be able to communicate with each other i.e. they find each other with some unique identifier (e.g. IP address). This is a very basic requirement and it is provided in even analogous PSTNs, IP or most communication networks.

Another important statement is that CMFS does not care about the access technology of the MN. This should be provided by the network. It is no matter wether the MN connected via WLAN or GSM or even MIPv4 specified access. The only thing is that the MN should be registered to a MAP in the CMFS what is considered as its connection point. Handover decision algorithms between the MAPs in our CMFS will be implemented in the MN and are discussed later.

The good thing in the requirements above that the CMFS protocol is totally independent of all the underlaying technologies no matter whether the communication goes over IP (BGP, OSPF, IS-IS), ATM (IISP, PNNI) in a GSM (TCAP) environment or anything else the important is that the nodes find each other.

2.2 The CMFS in the mobile node

The basic of the solution is that the MAs and MAPs in the network understand messages and commands from the MN. At first we show how the MN can find out the network structure and then we give an explicit specification of the commands i.e. the messages the MN sends to the network nodes.

2.2.1 Network Discovery - Logical Network

The first task for the MN to discover the network. It can send a message to any of the MAs it knows (the MN has to know at least one MA, the one that plays a Mobile IP Home Agent (HA)-like role). Also in most cases the MN is in a Foreign Network and thus attached to a MAP. Sending the message to the HA through the Network all the nodes the message passes by should reply with their ID (IP address if it is an IP network) to the MN so that it could calculate their logical position by the delay data. This is how the MN will build a Logical Network (LN) it can use as an input for its algorithm.

The solution we have chosen in our implementation is an IP based one and uses the IP traceroute packets. By using small TTL values which quickly expire, *traceroute* causes that the routers along a packet's normal delivery path generates automatically an ICMP Time Exceeded message. Measuring the delays the costs of using these links are determined.

The Logical Network structure for the MN must not be the same as the real network topology. Since the Logical Network is built up with measurements we think that it gives a better view of the real status of the network. For example if a node is down then it does not reply to the MN i.e. the MN thinks that it does not exist what is actually true from the MN point of view since the MN can communicate with operating nodes only.

2.2.2 Message structure and actions

The main element of our solution is that the MN orders the MAs to modify a kind of routing database they maintain. This database has entries like: "The MN can be reached via the MA_i or MAP_i"; so if a packet for the MN arrives to a given MA, it checks if there is an entry in its database and routes according to the rule it finds. If there is no such route than the packet has a correct destination on the mobility level so no routing is needed only what the underlaying network provides. The MA and MAP should be capable to register and delete new entries from the database upon the commands from the MN. Also they have to be able to forward such messages to each other (but this is again done easily on the underlaying network level.)

2.3 An implementation of the message format - Protocol

Let us show a possible CMFS implementation for specification of the above mentioned command structure, into an IPv4 network in the application layer. A CMFS message is carried in UDP packets thus uses the transport and the services from underlaying layers. We have chosen it instead of the TCP, like in other mobility solutions, because the TCP does not operates well with the radio interfaces. The TCP conceives the high bit error rate of the radio channel as congestion, and decrease the window size that ends in significant speed fall-off. For this reason the mobility applications generally use UDP to the communication.

The CMFS message structure follows strict rules as you can see on Figure 1. The header contains 4 fields of 1 byte elements: type, length, flags, number of actions, two 4 byte elements, an identification section and a variable length actions/informations field. Two different type of CMFS messages are differentiated, a *CMFS request* and a *CMFS reply* message. The length shows the full length of the CMFS packet included the header. Most of the bits of flag field helps the Network Discovery Procedure. One by one the bits mean the following:

- F - The first MA receiving this message sends a *CMFS Reply* and then set this bit to 0.
- S - If the message destination is the given MA and the S=1 then it sends a *CMFS Reply* (partial network discovery)
- L - For any message destination if this bit is set then the MA should send a *CMFS Reply* even though it only forwards the packet (full network discovery)
- C - This bit is examined if the node has to send a *CMFS Reply*. If this bit is set to 1 the MN expect capacity, and overload information from the MA in the *CMFS Reply* message.

The *number of action* field shows the amount of Action and Information contained in the packet payload. If the *destination* field matches the node identifier than it has to process the message and execute the actions in it. The *mobile current IP address* is needed for the Network Discovery Procedure so that the MA knows where to send the *CMFS reply* message. For this reason theis is an ptional field. The *identification* is a 64-bit number that uniquely identifies the CMFS message and is used to match *CMFS*

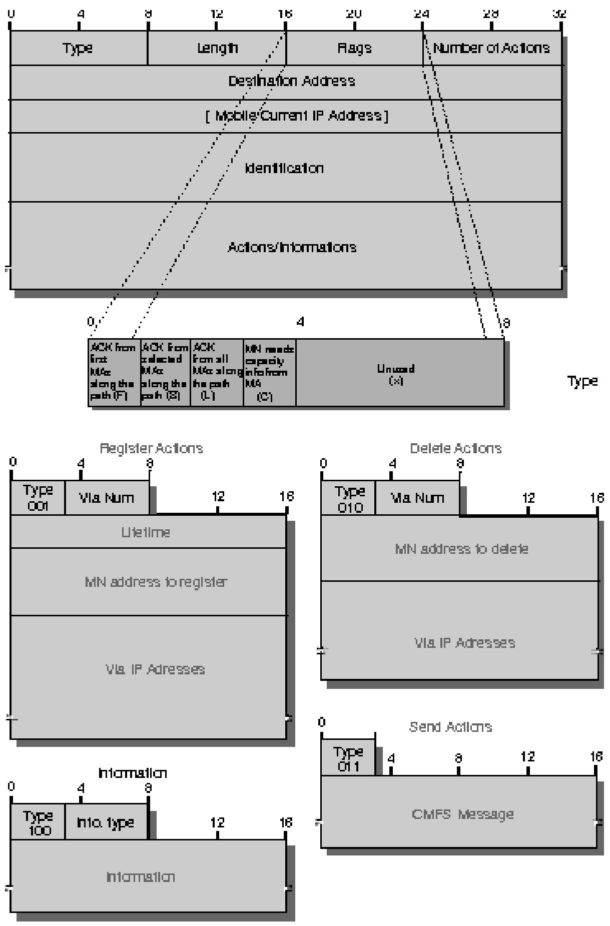

Figure 1: The CMFS message format

requests to *CMFS replies* and can explicitly provide protection against replay attacks. The *payload* of the CMFS message contains either the actions which have to be accomplished at the specific nodes or useful information, like capacity, for the network entities.

There are three different kind of actions. The first is the Register action that indicates a route registration in the given MA *via* the given destination *to* a specified target. The *lifetime* set by the mobile can assure the soft state data store at the MAs. Cleaning up the IP network is the problem of many mobility management systems. The obsolete registrations in the network nodes causes performance loss or even miss-routing of packets. Because of these it is very important to choose a proper *lifetime* for registrations. An optimal value can save the network from the unnecessary signaling also. 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 CMFS message.

3. THE MN BASED MANAGEMENT STRATEGIES

We have shortly introduced the new idea and explained the basics of its operation. In this section we will show possible strategies the MN can use to make the management work. We will present different approaches here and discuss them separately but note that a single MN can use multiple of these depending on its location for example. (Suppose that mostly we are working in the university and we spend most of the day in our room and in two labs thus stay under a set of agents. However, in the afternoon and in the morning we travel long distance passing through many MAPs. At home we have a router that is our HA. Intuitively we can think that the cellular approach is useful at the university and a tracking-like solution is the most efficient on the way home.)

One more thing we have to point out is that all the MNs in the world will see different networks and optimize the mobility for themselves. For this reason we believe that our solution has a better network resource utilization than any other classical one.

3.1 Mobility management strategies using CMFS, inspired by classical solutions

Here we present the versions of the basic Mobile IP protocols in the CMFS system.

3.1.1 *Personal Mobile IP – PMIP*

The operation of Personal Mobil IP is simple and easy and has been already discussed in our former papers [10]. Once the MN attaches to MAP it registers himself 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: MAP_i, Src: MN, Actions: Register MN to MAP_i via MN;
 [Dst: HA, Src: MAP_i, Actions: Register MN to HA via MAP_i
 ];
 [Dst: MAP_{i-1}, Src: MAP_i, Actions: Delete MN in MAP_{i-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).

The simple PMIP protocol operates alike MIP and has approximately the same capacity consumption as well as we will later see. We would like to point out that the MN has to maintain a Logical Network of 3 node 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 the what happens in case of a handover on Figure 2.

[Dst: MAP_i, Src: MN, Actions: Register MN to MAP_i via MN;
 [Dst: MAP_{i-1}, Src: MAP_i,
 Actions: Register MN in MAP_{i-1} via MAP_i;
 Delete MN in MAP_{i-1} via MN;
  [Dst: HA, Src: MAP_{i-1}, Actions: Register MN to HA via MAP_i;
  Delete MN in HA via MAP_{i-1};
   [Dst: MAP_{i-1}, Src: HA,
   Actions: Delete MN in MAP_{i-1} via MAP_i
   ]
  ]
 ]
].

Please find the performance analysis in Section 5.1.

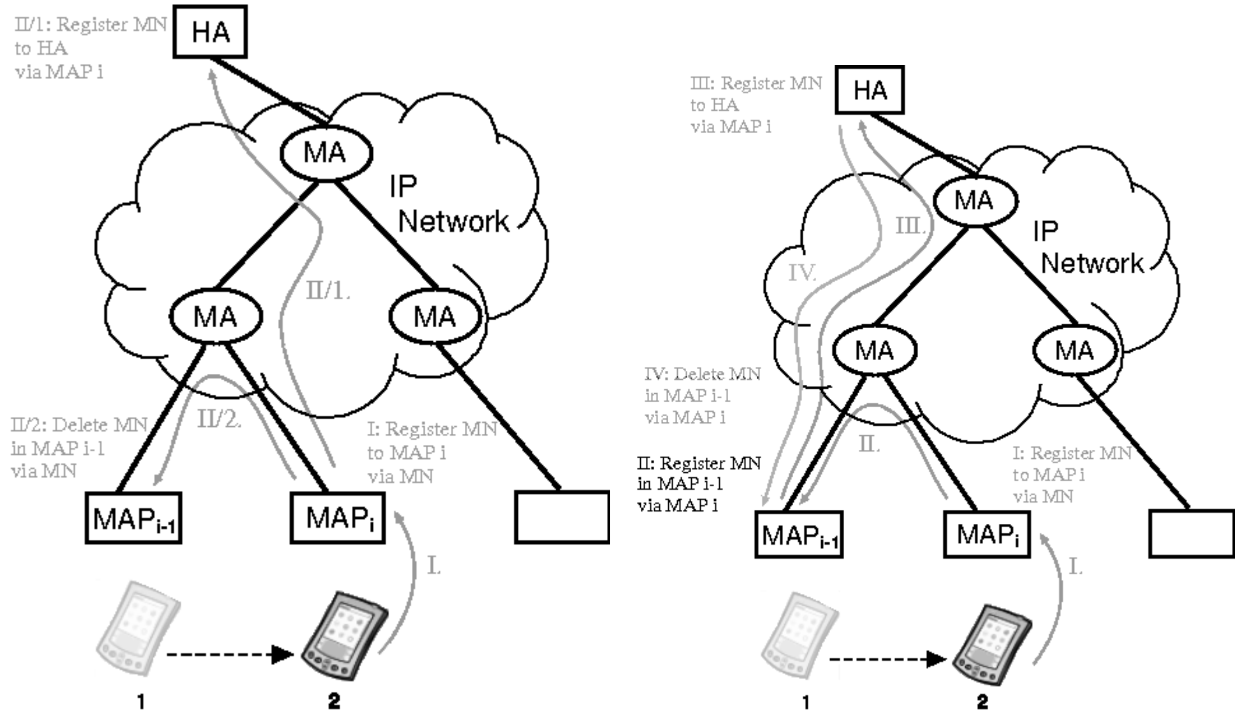

Figure 2: On the left hand side figure one can see the basic P-PMIP protocol, while the right hand side depicts the operation of the action-linearized Personal Mobile IP Mobility Management System (E-PMIP) with soft handover mechanism.

3.1.2 Personal Hierarchical Mobile IP – PHMIP

The operation of a HMIP micro-mobility (talking about 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 problem arise when talking about multiple layered hierarchical solutions. The MN has to make complex calculations 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 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.

3.1.3 Personal Tracking Mobile IP – PTMIP

A tracking-like (see Figure 3) 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 [4] or LTRACK [3] protocols.

The following message structure would then be used for the *tracking handover*.

[Dst: MAP_i, Src: MN, Actions: Register MN to MAP_i via MN;
[Dst: MAP_{i-1} //The former node//,
Src: MN, Actions: Register MN to MAP_{i-1} via MAP_i
]
]

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 to the HA (or to some hierarchy point in a more complex solution). One possible implementation of the *normal handover* would look like the following.

[Dst: MAP_i, Src: MN, Actions: Register MN to MAP_i via MN;
]
[Dst: MAP_{i-1} //The former node//,
Src: MN, Actions: Register MN to MAP_{i-1} via MAP_i
]

There are many proposed methods to decide between the two type of handovers. In our simulation we implemented a simple suboptimal solution when the MN registers back at every ith step.

3.1.4 Personal Cellular Mobil IP – PCMIP

Since the widespread use in GSM the cellular solutions became popular in most mobility applications. The ide is to avoid registrations when the MN moves within a given set of MAPs but then search it at each when it is paged. There

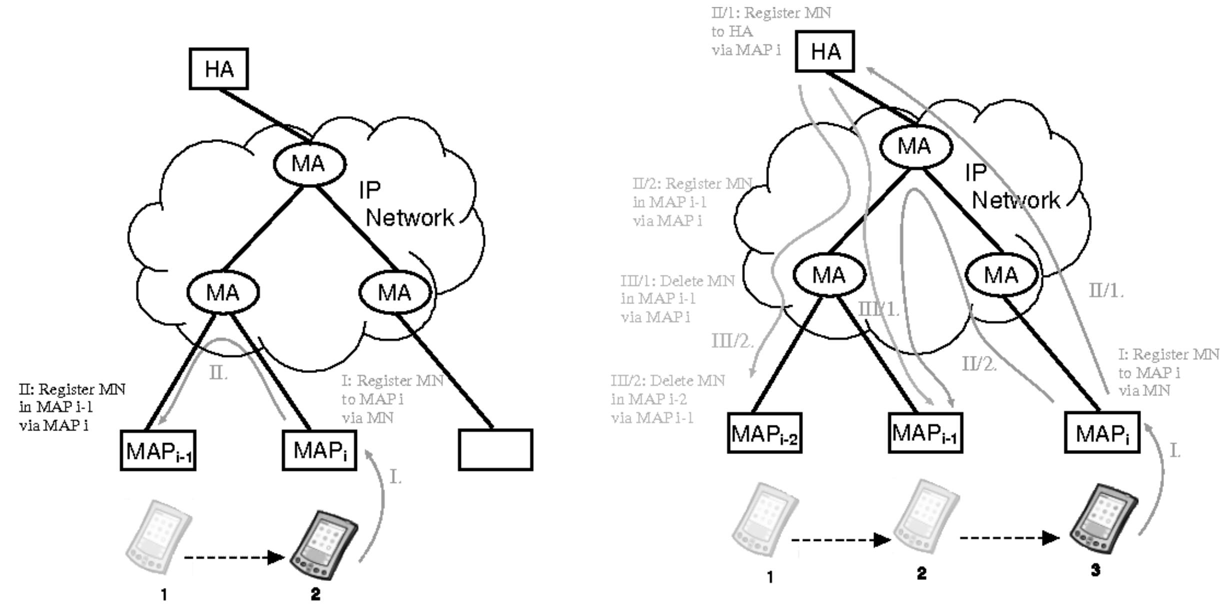

Figure 3: The operation of the Personal Tracking Mobile IP protocol. The *tracking handover* is depicted on the left hand side while the right is about the *normal handover*.

is a great 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. Also the MN it tells the IDs of the MAPs in the Paging Range (PR) to the leader of the PR so it will be aware who to broadcast the messages when the mobile is paged.

The following message tells to the one specific $\text{MAP}_{\text{leader}}$ (the leader) the MAPs belonging to that given PR: $\text{MAP}_i, \text{MAP}_{ii}, ...$:

[Dst: $\text{MAP}_{\text{leader}}$ //The leader of the paging area//,
Src: MN,
Actions: Register MN to $\text{MAP}_{\text{leader}}$ via MAP_i, MAP_{ii}, ... ,
[Dst: HA, Src: $\text{MAP}_{\text{leader}}$,
Actions: Register MN to HA via $\text{MAP}_{\text{leader}}$
]
]

And a message that can be used to register the PR of the MN at an upper level:

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

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. Consecutively the problem is NP hard for only one MN too. However, there are alternative solutions giving a solution what is good enough.

3.2 Some algorithms to set the parameters of the above protocols

At first we briefly present what king of information can the MN simply record to have a more sophisticated view of the Logical Network. Then we show how a possible cell forming mechanism would work.

3.2.1 Collecting Logical Network data

The algorithms we propose requires a set of extra data to be recorded by the MN. After switching on, the MN collects data from every network it attaches. Let $N_t = \max K, N$ denote the number of networks visited at time t.

The parameters we suggest to record:

- u_i: the cost of update from node i to the HA; can be determined from the traceroute times
- $u_{i,j}$: the cost of update from i to j; can be determined from the traceroute times
- d_i: the cost of delivery from the HA when the MN is paged;detto
- $d_{i,j}$: the cost of delivery from node i to node j;detto
- μ_i: the number of calls received at node i;
- $\lambda_{i,j}$: the number of movements from node i to node j,
- λ_j: the number of movements to node j.

These parameters will be used to model the network from the MN point of view, to decide the strategy and the actions. (This network model is very similar to the one used in our Mobility Management Framework (MMF) [**?**].)

One can extend the approach with recording for example Quality of Service (QoS) data or making reliability measurements, calculating costs of using each MAP (costs might be different if the IP mobility uses different service providers). However, in the present work we disregard these factors.

3.2.2 The optimal number of tracking handovers for PTMIP

The MN itself should decide whether to do *tracking handover* or *normal handover*. The optimal number of tracking handovers for the classical LTRACK solution using a Markov Chain model is calculated in [3]. Here we give a more simple alternative solution that gives an unique result for each MN depending also on their exact location.

Assume that the MN has just attached to a new MAP and has to decide upon the handovers. We assume that the best guess of the parameters of the consecutive MAP the MN will attach to is the same as the parameters of the actual MAP. (These assumption is very much like the martingale property assumption for stock value changes or call frequency changes in communication network. To further underpin it we should make measurements on real networks.) Furthermore if the parameters of the actual MAP are not known than it is assumed to be the same as the previous etc. (If only the MN moves to a MAP for which it have measurements from the past then of course it can use those for the calculation.)

It is worth to make a *normal handover* only if its expected cost is less than the expected cost of the tracking handover. All the cost has to be calculated considering possible deliveries too.

The expected cost with the *tracking handover* (at node i):

$$u_{i,\text{prev}} + \frac{\mu_i}{\mu_i + \mu_j} D_i,$$

where $D_i = d_1 + d_{1,2} + ... + d_{\text{prev},i}$ is the total cost of delivery through the chain of tracking points. The expected cost of a *normal handover*:

$$u_i + \frac{\mu_i}{\mu_i + \mu_j} d_i.$$

After the costs recorded the decision is easy to make.

3.2.3 Cell forming algorithm for PCMIP

We give an algorithm where, based upon a quasi-optimal location area forming algorithm (the problem of optimal cell forming is NP hard), the MN will configure the network to a spacial CIP or TRACKING-like model [12].

With this method the MN is able to operate with almost the optimal cost PR-s and signalling messages are saved too. The tradeoff is that the MN has to maintain its database, calculate the ideal network for itself and also the leaders of the PAs have to maintain a database of the attached mobiles. This shouldn't be too much since different MNs might chose different PR leaders and thus the database and the work is well distributed.

When the MN attaches to a node that is not in its database yet it has to put the new MAP into a PR immediately and notify the leader or even can off-line reconfigure the network.

The solution briefly is the following. The update and registration costs are recorded in each mobile individually for each network node. Then it is decided whether it is worth to merge a cell to another and form a new Paging Range. From now on let us use the word cell what is understood as a common word for node (MA) or Paging Range and also the atomic logical entity of the algorithm.

At first we define the specific cost cost of maintaining a cell, namely: i.

$$c_i = \mu_i d_i + \sum_{\forall j} \lambda_{j,i} u_i = \mu_i d_i + \lambda_i u_i$$

Now we define the extra cost of incorporating a cell into another:

$$c_{i,j} = \mu_j(d_i + d_{i,j}) + (\lambda_j - \lambda_{i,j})(u_{j,i} + u_i) - (\lambda_i - \lambda_{j,i})u_i$$

Speaking in words, the extra cost when the cell j is incorporated to cell i is the incoming call rate (μ_j) multiplied by the cost of the new delivery procedure through the node i: $(d_i + d_{i,j})$, plus the cost of the updates when the new cell is reached via the incorporated cell j: $(u_{i,j} + u_i)$, multiplied with this event's frequency: $(\lambda_j - \lambda_{i,j})$, where $\lambda_j := \sum_{\forall k} \lambda_{k,j}$. On the other hand, we also save the cost of moving into cell i from node j that should be multiplied by the frequency of this event: $(\lambda_{j,i})$.

The cost is defined if we want to merge only cell j. Similarly we can define it in the case if we want to merge an $\mathcal{M}$ subset of all the cells:

$$c_{i,\mathcal{M}} = \sum_{m \in \mathcal{M}} \mu_m(d_i + d_{i,m}) + \sum_{n \in N} \sum m \in \mathcal{M} \lambda_{n,m}(u_{m,i} + u_i) \quad (1)$$

Where $\mathcal{M}$ is the set of nodes we want to merge and N is the set of the not merged ones. Also $M \cup N$ is the set of all the nodes examined (neighbors). The algorithm goes as follows: Initially we start from a sorted list of costs of maintaining a cell, and try to merge the neighboring ones with the following rule:

1. Take the first element (node or cell with the maximal c: i);
2. For all M subsets of its neighbors $M \cup N$ (including the trivial ones):
   (a) Take the subset of the maximal cost of incorporation.
   (b) If $\sum_{j \in J} \lambda_j u_j + \mu_j d_j < \sum_{j \in J} c_{i,j}$ i.e. the cost of merging is higher than the cost of maintaining separate cells THEN "DO NOT MERGE" ELSE "MERGE" and recalculate the cost of the new cell. At first some notations and temporary variables:
      - $m \in M$: neighbors Merged + initial node; all the nodes in the new cell
      - $n \in N$: neighbors of the New entity, or all the nodes which were not merged and are not in the new cell
      - $p_m = \frac{\sum_{n \in N} \lambda_{n,m}}{\sum_{n in N} \sum_{l \in M} \lambda_{n,l}}$:the probability that we arrived to the merged node m from outside of the new cell.
      - $\overline{p_m} = \frac{\sum_{n \in N} \lambda_{m,n}}{\sum_{l \in M} \sum_{n \in N} \lambda_{l,n}}$: the probability that we left the new cell from the merged node m.

      Then recalculating each cost:

      - $c_{new} = \frac{c_i + \sum_{m \in M} c_{i,m}}{c_i + \sum_{m \in M} c_m}$ because the c_i is a relative cost;
      - $\lambda_{new,n} = \lambda_{i,n} + \sum_{m \in M} \lambda_{m,n}$;
      - $\lambda_{n,new} = \lambda_{n,i} + \sum_{m \in M} \lambda_{n,m}$ (the frequencies are simply summed);
      - $\mu_{new} = \mu_i + \sum_{m \in M} \mu_m$ this is trivial: the chance of receiving a call in the new cell is the chance of receiving a call in each subcell of it;

- $u_{new} = u_i + \sum_{m\in M}\sum_{n\in N}\frac{\lambda_{n,m}}{\sum_{n\in N}\lambda_{n,m}}(u_{m,i} + u_i)$ here we should see, that the update cost in the new, "supercell" is the update cost from cell i plus the update costs from the merged cells, what to which subcell we attach to. So we should sum up for every merged neighbor $\sum_{m\in M}$ and then for all the not merged neighbors of each $\sum_{n\in N}\lambda_{n,m}(u_{m,i} + u_i)$;
- $u_{n,new} = \sum_{m\in M}\frac{\lambda_{n,m}}{\sum_{n\in N}\lambda_{n,m}}u_{n,m}$ We sum up the $u_{n,m}$s, but each has to be normalized with the probability of using it;
- $d_{new} = d_i + \sum_{m\in M}\mu_m(d_i + d_{i,m})$ The fact is that this works only when one of its 1-distance neighbors are merged. If multiple one is merged then the;
- $d_{new,n} = \sum_{m\in M}\overline{p_m}d_{m,n}$ what is the weighted value of the delivery cost i.e. sum for all m the cost of delivery from m multiplied by its probability.

Then sort the new cells (insert the new cell into the right position) and start it again.

The algorithm ends when there is no benefit of merging any cell with any other.

Note that there are two things why this is not the optimal solution. The first is because the leader of the PR is selected without any cost check and secondly because we propose to examine only the neighbors of a PR. (It is mathematically possible that it does not worth to incorporate a neighboring while it worths to incorporate another, far away cell.) However, we still gave a solution good enough.

4. SIMULATION AND NUMERICAL RESULTS

We have made a simulation to show at first that or proposal actually works and secondly to compare it to existing technologies. The simulation was written in the open source OMNet++ [14] using C++ language.

It is essential to point out that the simulation is written such a way that it can easily cooperate with the one presented by us for classical mobility solutions like MIP, CIP, HMIP, LTRACK, etc. We use this to compare the methods. For details see. [12].

4.1 The structure of the CMFS simulation program

The simulation consists of two main modules namely *MN* and *MA* and some other simple components that are needed to model the operation environment (see Figure 4). The two main modules has similar internal structure. Both has a *DataSender* and a *DataReceiver* 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 *NodeCore* 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 to receive and analyze it. The number and size of packets, the frequency of data sending 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 also can be extended to record other QoS parameters like delay or jitter too.

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

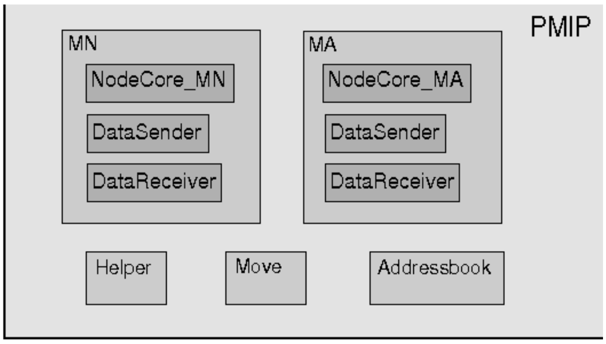

Figure 4: The component structure of the simulation of CMFS written in OMNet++.

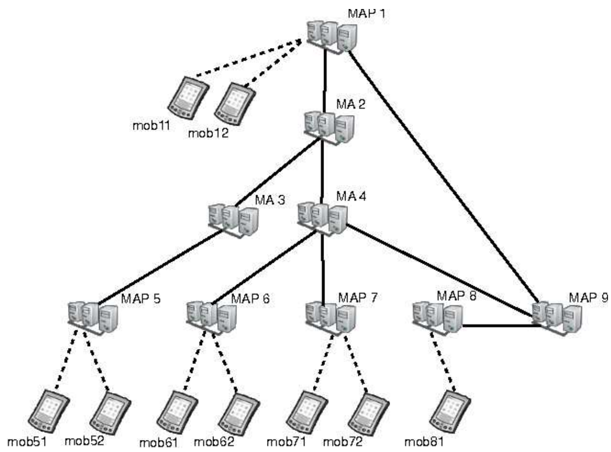

Figure 5: The component structure of the simulation of CMFS written in OMNet++.

4.2 The tested scenario

We have constructed a virtual test environment consisting of 9 MAs and 9 MNs with the initial MN distribution depicted on Figure 5.

5. COMPARISON AND ANALYSIS OF THE PROTOCOLS AND SOME NUMERICAL RESULTS

In this section we grabbed three main aspects of the new solution to analyze it. At first we show how its costs relates to the simple MIPv4 solution purely collecting network signalling data and examining the whole from QoS point of

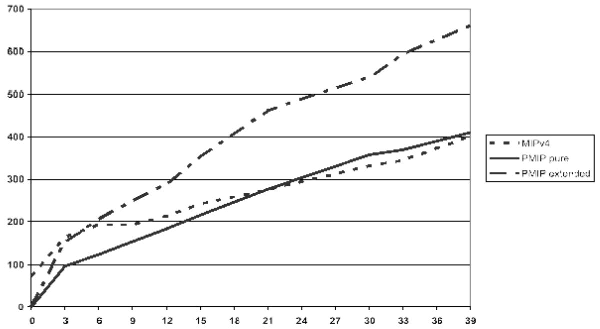

Figure 6: Comparison of the three mobility Management Systems presented. The horizontal axis shows the number of handovers between two arrived calls while the number of bytes transmitted on the network by each protocol is presented on the vertical axis.

view. Then we give some numerical results on the performance of a more complex solution, PTMIP. Finally we discuss some conceptional issues that reveal some fundamental differences between our new protocol and the MIP versus its existing extensions.

5.1 Comparing the basic approaches

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

The performance of the protocols is depicted on Figure 6. 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 falls 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 the 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 need *Agent Advertisement* messages to maintain connectivity while in the client based system can rely on lower layers.

What we can conclude is that the basic solutions work with approximately the same cost. However, E-PMIP shows that it is possible to improve the performance while not changing the protocol at all (only on the MN side).

5.1.1 Comparison with a QoS-like cost function

We have already presented a figure to compare the approaches from the signalling load point of view. Of course it is easy to define a kind of QoS cost function too and examine the performance in the mirror of that. We construct a very simple kind of QoS cost function.

We can assume that the *MN can attach to only one MAP at a time.* If not than both protocols could benefit from that the same way that is the MN can receive duplicated messages while changing the MAP. The second assumption that the *signalling and the data is transmitted with the same speed between MAs.* This is rather reasonable or at least if there is a difference because of IP packet prioritization than the same method is expected to be applied for the classical and the new solutions. As a third assumption we say that because of the second one, *the transmission speed is in the same linear relation to the logical distance of MAs for both solutions.* We introduce a network and mobility related parameter g_T that is the distance from the previous MA the MN has attached and the new one. Let us note the average distance from the HA with m. Similarly g_C will denote the logical distance to the FA or MAP the MN attaches. The $t_{\mathrm{MA}}^{\mathrm{Protocol}}$ and $t_{\mathrm{HA}}^{\mathrm{Protocol}}$ will denote the average time of processing Protocol messages at the specified nodes.

We measure QoS cost in the relative number of packets to wrong direction for each protocol. We assume that this is a linear function of time thus a linear function of the logical distance and processing time. This linear function can be set to the same for all the protocols and will be chosen to Identity for simplicity. (Of course this comparison can be further elaborated if intended.) Now let us see the QoS cost:

$$\begin{aligned} C^Q_{\mathrm{MIP}} &= g_C + t^{\mathrm{MIP}}_{\mathrm{MA}} + m t^{\mathrm{MIP}}_{\mathrm{HA}} \\ C^Q_{\text{P-PMIP}} &= g_C + t^{\text{P-PMIP}}_{\mathrm{MA}} + m t^{\text{P-PMIP}}_{\mathrm{HA}} \qquad (2) \\ C^Q_{\text{E-PMIP}} &= g_C + t^{\text{E-PMIP}}_{\mathrm{MA}} + g_T t^{\text{E-PMIP}}_{\mathrm{MA}} \end{aligned} \qquad (3)$$

and thus we can derive the QoS cost relation between the protocols:

$$C^Q_{\mathrm{MIP}} - C^Q_{\text{P-PMIP}} = (t^{\mathrm{MIP}}_{\mathrm{MA}} + t^{\mathrm{MIP}}_{\mathrm{HA}}) - (t^{\text{P-PMIP}}_{\mathrm{MA}} + t^{\text{P-PMIP}}_{\mathrm{HA}}), \qquad (4)$$

that is rather hard to handle since depends very much on the implementation and the working nodes. But it is easier to see the clear difference between the MIP and E-PMIP if we assume that all processing times are equal to t_{PROC}:

$$C^Q_{\mathrm{MIP}} - C^Q_{\text{E-PMIP}} = m - g_T. \qquad (5)$$

This is the time difference in traffic disturbance. It is somehow logical to see that in most of the times $m \geq g_T$ since the HA is often farer from the new MA than the old MA.

5.2 Numerical results for complex mobility managements using CMFS

We have run some simulation to provide results on the performance of the tracking-like solution PTMIP. In this case we speak about such a tracking when the messages are sent through the links of the network not through the air interface.

On Figure 7 one can see that in this case the most *tracking handovers* performed and thus the most *normal handovers* avoided significantly decreases the signalling cost for the protocol.

5.3 Differences in the network implementation

In this Section we try to focus on the benefits of the fact that the client based mobility management does not need a pre-built network topology. Both in the simulation and

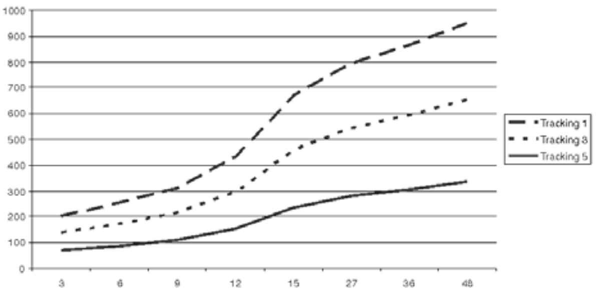

Figure 7: This figure depicts the perfromance of three tracking-like approaches namely PTMIP with 1, 3, 5 tracking handovers. Note that for this simulations a couple of additional links were inserted to the network.

in the case of the above calculations we had an assumption that the same nodes run mobility management system in the network for the MIPv4 and the PMIP protocols. We say that this is not necessarily true.

If we implement Mobile IP to a system, we have to install HA and FA functionality into all the networks we want to use for the communication. If a new FA is installed to the system it can automatically co-operate with all the different HA in the network according to the RFC. This is the same in our proposal and there is no need for a complex network structure.

The differences appear if we want to extend the algorithms. The classical MIPv4 enhancements such as HMIP [2], TeleMIP [7], DHMIP [4], CMIP [1], Hierarchical Paging [8], LTRACK [3], etc need to have a pre-built network structure i.e. the MAs in a HMIP has to be aware of the hierarchy structure. It is a great advantage that this is not needed in our case since the MN itself can administrate the Logical Network for itself. This also allows to build different hierarchical or cellular structures for each node in an optimal way to provide QoS or ensure cheap operation (low signalling or processing cost).

One can see that any node can join and leave the network at anytime just like in the MIP case but meanwhile a more complex mobility management can be applied. To discuss the cases when an FA or an MA breaks down or joins the network is part of our upcoming work.

6. CONCLUSION

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

7. ACKNOWLEDGEMENTS

This work was supported by the Mobile Innovation Center, Hungary and partially by the High Speed Network Laboratory, Hungary.

8. REFERENCES

[1] A. T. Campbell, J. Gomez and A. G. Valkó: *An Overview of Cellural IP*, IEEE. pp 29–34. 1999.

[2] C. Castelluccia: *A Hierarchical Mobile IP Proposal*, Inria Technical Report. .1998

[3] Kovács, B., Szalay M. and Imre, S.:Modelling and Quantitative Analysis of LTRACK - A Novel Mobility Management Algorithm. Mobile Information Systems, Issue: Volume 2, Number 1/2006 Pages: 21 - 50. (2006)

[4] Ma, W. and Fang, Y.: Dynamic Hierarchical Mobility Management Strategy for Mobile IP Networks. IEEE Journal of Selected Areas In Communications. (2004)

[5] Perkins, C. E.:Mobile IP. IEEE Communications Magazine. (1997)

[6] Ramjee, R., La Porta, T., Thuel, S., Varadhan, K. and Salgarelli, L.: A Hierarchical Mobile IP Proposal. Inria Technical Report. (1998)

[7] S. Das, A., Misra, P. Agraval & S. K. Das, TeleMIP: Telecommunications-Enhanced Mobile IP Architecture for Fast Intradomain Mobility, 2000, IEEE Personal Communications, 50-58.

[8] Szalay, M., Imre, S., *Hierarchical Paging- Efficient Location Management*, 2007, Fourth European Conference on Universal Multiservice Networks (ECUMN'07), 301-310.

[9] Vilmos Simon, Sándor Imre: Location Area Design Algorithms for Minimizing Signalling Costs in Mobile Networks, International Journal of Business Data Communications and Networking (IJBDCN), (2007)

[10] P. Fülöp, B. Kovács, S. Imre, "A Client-driven Mobility Frame System - Mobility Management From a New Point of View"

[11] P. Fülöp, B. Kovács, S. Imre,"A Client-driven Mobility Frame System - Implementation to IP Networks"

[12] P. Fülöp, B. Kovács, S. Imre,"Numerical analysis of mobility management methods" Info-communications-technology, 1945 volume LXII 2007/7, Pages: 32-38, Budapest

[13] Wolfram Research Inc.: *Mathematica*. http://www.wolfram.com/.

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