COMPLEX MOBILITY MODELLING ALGORITHMS
AND APPLICATIONS

Fülöp Péter

Ph.D. Dissertation

Supervised by

Dr. Sándor Imre

Budapest, Hungary

2015.
ACKNOWLEDGEMENTS

I would like to thank my supervisors, Dr. Imre Sándor and Dr. Szabó Sándor, for their valuable guidance and patience. They have given me stimulating ideas and constructive comments on my research.

I want to thank my colleagues Kovács Benedek and Szálka Tamás for their precious comments in all the discussions.

Last but not the least special thanks go to my wife, Hoferik Anita for her constant encouragement and support throughout my research period.
Abstract

Information mobility has become one of the most important achievements of the modern world. The number of subscribers and also the network traffic is increasing in the mobile cellular networks. Smartphones with more network interfaces keep spreading, while on the other hand the network operators implement technological improvements and standards to be able to cope with increased data transfer needs and also introduce sophisticated services in the market competition. As a result the operators are facing even more complex operational and management tasks.

The cell sizes keep getting smaller and change dynamically, which means signaling traffic increase. Mobile systems provide enough bandwidth for present mobile multimedia applications. However such applications remain sensitive to the sudden degradation of QoS parameters. In this changing environment efficient and adaptive user localization and call transfer is needed to solve the mobility management tasks. It is a challenging task to create a completely safe mobile environment due to the open radio interface and the vulnerability of the mobile devices. A prospective, alternative possibility is to introduce such procedures and mechanisms that enables the systems to reconfigure, heal or install themselves according to the changing environment. Well-structured mobility management algorithms can support for example the call admission control, reconfiguration of the cell coverage, detection of the fault of the radio interface or a change in the location management strategy.

In my PhD Dissertation I introduce my mobility management model, which is able to compare the main mobility management strategies in the literature from signaling, processing in the nodes, air interface usage and security point of view. For the different approaches I defined cost functions, which is able to be identified the mobility management that gives the best solution in different given network scenarios and which aspect of resources could be a bottleneck in each case.

I proposed a classification for for the discrete-time Markov movement models which states express the staying in an area. The solution is able to compare and to analyze the efficiency of Markov movement models in different network scenarios. Based on the classification experiences I created a general, discrete-time Markov Movement model Creator Framework, and the $n+2$ state $Mn$ model.

In the third part of the Dissertation I introduced create a self-organizing location management environment, where every mobile node manages their location update strategies for itself not the network. Using this every mobile or group of mobiles is able to select the mobility strategy which cause the minimum signaling cost from the network point of view.
Table of Contents

1 Introduction ............................................................................................................................. 5
  1.1 Terminology ......................................................................................................................... 6
  1.2 Research Objectives ............................................................................................................. 7
  1.3 Research Methodology ......................................................................................................... 7
  1.4 Structure .............................................................................................................................. 8

2 Enhanced framework for modeling mobility management strategies ........................................ 9
  2.1 Mobility Management Model - MMM .................................................................................. 10
    2.1.1 Basic notations and assumptions ................................................................................. 11
    2.1.2 Network Graph and Node Mobility Parameters .......................................................... 12
      2.1.2.1 Modeling the network ............................................................................................ 13
      2.1.2.2 Deriving parameters of a given network ................................................................. 13
      2.1.2.3 Modeling the mobile node ...................................................................................... 16
    2.1.3 Cost functions methodology and cost constants ............................................................ 16
      2.1.3.1 Link related constants ............................................................................................ 16
      2.1.3.2 Node related constants ........................................................................................... 16
      2.1.3.3 Mobile equipment connection related constants ..................................................... 17
      2.1.3.4 Security cost class .................................................................................................. 17
      2.1.3.5 Example of cost diversification .............................................................................. 18
  2.2 The main management strategy classes .............................................................................. 18
    2.2.1 Centralized approaches ................................................................................................. 19
    2.2.2 Hierarchical solutions .................................................................................................. 20
    2.2.3 Tracking-like solutions ................................................................................................ 21
      2.2.3.1 Wireless tracking ................................................................................................... 22
      2.2.3.2 Wired tracking ....................................................................................................... 23
    2.2.4 Cellular-like solutions ................................................................................................. 24
      2.2.4.1 Standard Cellular-like solutions ............................................................................ 25
      2.2.4.2 Manet-like solutions ............................................................................................. 25
      2.2.4.3 Hierarchical Paging-like solutions ....................................................................... 26
  2.3 Compare different approaches ............................................................................................ 27
    2.3.1 General differences between management approaches ................................................. 27
    2.3.2 Technology tendency via cost constants and simulation ............................................... 31
    2.3.3 How to use the model for vertical handover decisions .................................................. 32
    2.3.4 Compare the battery consumption of different approaches ......................................... 34

2.4 Overview .............................................................................................................................. 36

3 General mobility modeling and location prediction based on Markovian approach – MMCS-Markov Movement model Creator System ................................................................. 37
  3.1 Mobility models in the literature ......................................................................................... 37
    3.1.1 Mobility models ............................................................................................................ 37
    3.1.2 Group mobility models ................................................................................................ 40
    3.1.3 User-distribution prediction models ............................................................................. 40
  3.2 Classifying Markov mobility models ................................................................................... 41
    3.2.1 Deriving user movement related network parameter of given network for MMM ........ 41
    3.2.2 Markov model ............................................................................................................. 44
    3.2.3 Classification ............................................................................................................... 44
    3.2.4 User-centralized Markov models (UCM) ...................................................................... 45
1 Introduction

Information mobility has become one of the most important achievements of the modern world. The number of subscribers and also the network traffic is increasing in the mobile cellular networks. The popularity of the mobile communication is a result of the combination of different factors. Smartphones with more network interfaces keep spreading, while on the other hand the network operators implement technological improvements and standards (HSXPA-High Speed Packet Data, LTE (Long Term Evolution) to be able to cope with increased data transfer needs and also introduce sophisticated services in the market competition. As a result the operators are facing even more complex operational and management tasks.

The cell sizes keep getting smaller and change dynamically; the reuse of the available radio channels is increasing also. In this changing environment efficient and adaptive user localization and call transfer is needed to solve the mobility management tasks.

The enhanced radio techniques applied in mobile systems provide enough bandwidth for present mobile multimedia applications. However such applications remain sensitive to the sudden degradation of QoS parameters. To avoid such degradation two important issues should be considered; proper dimensioning of physical network resources is essential and by using Call Admission Control (CAC) the operator can limit the number of newly accepted connections, thus achieve traffic shaping [14]. Accurate, yet simple mobility models are required to address these issues.

In recent years people increasingly rely on wireless devices in their daily life for very sensitive tasks such as purchases handling and bank transactions. Although there are several available protocols that guarantee our safety in the wireless environment, it is still a challenging task to create a completely safe mobile environment due to the open radio interface and the vulnerability of the mobile devices. Anomaly-based detection as part of the detection-based techniques creates normal profiles of system states or user behaviors, stores and periodically compares them with the current activities. If significant deviation is detected, the network system raises an alarm. However a user profile is very difficult to build up, it could significantly increase the safety of the mobile networks [15][16][17]. Mobility models and location prediction could take significant parts of creating user profiles as well.

Nowadays there are multiple access networks available at the same time in one geographical location. Besides there can be overlapping between the cells with the concrete access technology. Hierarchical cells and the usage of “umbrella cell” are good examples, when covering more smaller cells with a greater one. The reason behind is serving the fast users and to avoid faults resulting from frequent call transfers. The movement models support the process of possible vertical call transfers at these locations.

Although there are other solutions there is still a need for IP mobility since IP is the most widespread protocol and the new, 4G mobile solution (LTE – Long Term Evolution) is All-IP network as well. The communicating equipments are identified with their permanent IP address and the communication is done on IP networks, hence a well developed IP based mobility strategy is essential for the network providers. Many works have discussed the problem of managing the movement of the clients, since the IP 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) of the call/data delivery and optimality of the location update. Naturally, this can not be resolved, but I transform it into another dimension: from network level to mobile node level.

These were just a couple examples from the numerous challenging tasks in the mobile networks and to be able to cope the operators need appropriate solutions and highly qualified, experienced engineers. A prospective, alternative possibility is to introduce such procedures and mechanisms that enables the systems to reconfigure, heal or install themselves according to the changing environment [18]. These systems shall be capable of modifying their own behavior and adapt to environmental changes based on performance measures. However the initial research area is far from being exhausted it can be seen that well-structured mobility management algorithms can support for example the call admission control, reconfiguration of the cell coverage in 4G LTE network, detection of the fault of the radio interface or a change in the location management strategy.

1.1 Terminology

Upon setting up a wireless network service access points are located over the targeted geographical locations. These service access points are called base stations. The base stations’ radio interfaces cover a specific area, which is called a cell. The moving user end devices, called cell phones, can communicate with the network – and with each other - through the base stations. The mobile devices are usually moving and therefore change their access points. If during the movement they have an active connection/call the process is called a handover. The previously described functionality has to be managed by the mobility management service; one of this service’s main functions is the location management. The location management means the recording/registering the mobile’s location in the network with a predefined precision/punctuality. This is necessary so that the network is able to route any incoming call or data traffic towards the mobile device.

In my Dissertation, during the modeling the mobility is treated generally, with no respect to the exact, applied technology. Accordingly I use the following network entities:

- **Mobile Node (MN):** moving node, which is communicating with another mobile or fixed base station.
- **Mobility Access Point (MAP):** the only type of fixed entity in the network to which a mobile node can connect and communicate through. It does not necessary match an access point in the real network. It can mean a set of access points if a mobile can be in connection with more base stations, or it can be a network (for example a MAP can represent the whole RAN – Radio Access Network).
- **Mobile Agent (MA):** fixed node, runs the mobility management algorithm, but no mobile can connect to it. A hierarchy node is such for example.
- **Central or Home Agent (HA):** a special, central MA, which always has an exact, or rough/approximate information on the whereabouts/location of an MN.
- **Other node or network entity (ON):** such node in the network, that has no function related to mobility management. Fix communication partner, routers and other network elements belong here.
Network (N): the communication channel (wired or wireless) connecting the entities, which is represented by a graph. The junctions of the graphs represent the above mentioned network entities, the edges of the graphs represent the connections between the entities.

1.2 Research Objectives

In cellular mobile network the number of users and the amount of transferred data increased significantly. Due to this fact the network operators are facing heavy problems. My research objective with user movement modeling and mobility management strategy analyzing is to support designing and maintaining the next generation mobility networks and to introduce new approach of adaptive, location management algorithm.

My goals are:

- to investigate and compare location management, a part of the mobility management, strategies with respect to their need of signaling and processing resources on the backbone network, load/utilization of the air interface and the cost of the security.

- to classify and investigate Markov movement models applied in cellular mobile networks. Based on my research to create an optimal model and develop general Markov Movement Model Creator System (MMCS), which generates an optimal (in the number of states and the prediction accuracy) Markov model using the network and mobility input parameters.

- to create a self-organizing location management environment, where every mobile node manages their location update strategies for itself instead of the network (Client-driven Mobility Frame System – CMFS). The network only provides basic services for mobile entities: connectivity and administration.

Accomplishing my research goals we get one step closer to a self-organizing, intelligent mobile network, where the mobile node always uses the lowest-cost mobility management strategy and can be tracked/followed by an optimal movement model.

1.3 Research Methodology

In order to prove the efficiency of the management strategies and movement models I used two classical approaches: analytical considerations and simulations.

First the mathematical model had to be constructed for the analytical consideration. I used graphs to model the network and Markov chains for position management solutions and movement models. In the course of comparing the location management strategies I applied a mobility model created from an existing model in the literature according to my research goals. The extension of this model is used during movement model classification.

I prepared a Mathematica [8] application to confirm the analytical considerations and developed a simulation environment. The simulations were written in the open source, widely known, discrete time OMNet++ [52] using C++ language. Using these the results are comparable to the approaches in the literature and the effect of different network configurations can be measured.
1.4 Structure

The theses are organized around three concepts:

I. Enhanced framework for modeling mobility management strategies

II. General mobility modeling and location prediction based on Markovian approach – Markov Movement model Creator System

III. Mobility from a brand new point of view: the Client-driven Mobility Frame System

The ‘mobility modeling’ in the Dissertation means two well separated categories. The first is the ‘movement modeling’, which aim is to predict the future position of mobile entities, or in other way, to determine the number of users in a well determined area. The second is the ‘mobility management modeling’, that ensures permanent availability in the network for mobile nodes during their movements.

This Dissertation is structured as follows:

In Chapter 2 (Thesis 1.) the enhanced model for the network and the mobile node with its mobility parameters are introduced. This is followed with the definitions of the complex cost functions for existing approaches. One can see the comparisons, figures of the numerical results and the conclusion in chapter 2.3.

The next Chapter (Theses 2-5.) contains a brief introduction of the movement models. I extend the network model and propose the classifying of Markov approaches, give examples from the literature, and introduce new models to fill the gaps. This is followed by a general methodology in order to determine the optimal Markov model for a network. If it could be predicted or modeled the user movement, than the next step to chose the most optimal mobility for itself.

In Chapter 4 (Theses 6-8.) Client-based Mobility Frame System is presented, and then I give mobility applications what is the Mobility Frame Systems itself. Later I present smart mobility management strategies and compare them to classical approaches. The implementation issues over IPv4 and IPv6 are discussed and finally simulation and numerical results are given.
2 Enhanced framework for modeling mobility management strategies

In the first mobility protocol designs, the main scope was to create a well-functioning mobility. For example, the Global System for Mobile Communication (GSM) [53] network uses a Cellular approach to save bandwidth on the air interface but does not really focus on the problem of signaling load on the wired serving network. The GSM uses static location update strategy, namely zone-based scheme [54], where the coverage area is divided into well defined group of cells, called Location Area (LA). When a mobile in idle state crosses a boundary of two LA, then send location update information to the network. Obviously a mobile with active call sends an update message every time it moves to a new cell.

The current 3G mobile network, UMTS (Universal Mobile Telecommunications System) [55] use static location update solutions combined with different paging strategies as well. The cells are grouped into Routing Areas (RAs) and RAs grouped into UTRAN (UMTS Terrestrial Radio Access Network) registration areas (URAs) [56]. Therefore UMTS applies three-level location management strategy, so a mobile is tracked at cell level during packet transmission, at the URA level during the idle period of an ongoing session, and at the RA level without any communication session.

In the 4G LTE network the cells are grouped only into Tracking Areas (TAs) because the network is fully IP based and it has no exact connection to the Core Switch. Tracking areas are equivalent to Routing Areas in UMTS.

The concept of NGN (Next Generation Networks) was realized in IMS (IP Multimedia Subsystem) by standardization institutes. In this built dream the common backbone network deliver all the required signal- and data transfer of all occurring telecommunication and internet service without any complexities. In this concept the location management is also an important key task.

The most widely known, first IP solution was the Mobile IPv4 (MIP) [30], which was developed for IPv6 (MIPv6) [47] [48] as well. Later on newer, improved versions of the mobile IP appeared, including the Hierarchical Mobile IP (HMIP) [33] [49] and the Dynamical Hierarchical Mobile IP (DHMIP) [31]. The basic idea of these protocols is that instead of the global management system a local system is used to minimize the network signaling traffic. Certainly other solutions were published besides Mobile IP, like Cellular IP (CIP) [32] which supports fast, local mobility or Host Identity Protocol(HIP) [37], which separates the IP address’s global identifier and localizing function. The Location Tracking (LTRACK) [34] developed for low call ratio and the hierarchic version of the paging solution, Hierarchical Paging (HPAGE) [43], known from CIP are also improvements of the earlier published mobility protocols [38] [39]. Application level solutions like Session Initiation Protocol (SIP) [40] were also born. Papers describing the integration of the existing protocols were also published, like the SIP and MIP two layered mobility [44], which provides fast hand-off and results in decreased global signaling traffic. The previously introduced Home Agent Architecture’s adequate implementation [46] decreases the time of location update and the time of tunneling.

The Wireless Local Area Networks (WLAN) is constructed similarly to the original Local Area Networks (LAN) and provides mobility only within the radio interface and uses Dynamic
Host Configuration Protocol (DHCP). Future protocols might use different media and technological background to provide mobility.

The advantage is that I do not focus on a selected technology not even on a given network generation but discuss mobility in general within the modern computer and telecommunication networking technologies. To describe the environment I focus on the bottlenecks such as the air interface, bandwidth on the network and processing load on the network nodes and as one of the most significant properties: complexity involving scalability.

I compare selected mobility approaches and show how the network properties affect the usability of each. The aim is to find the suitable one for different scenarios and to give guidelines how to construct the network for a protocol or a protocol to the network. To generalize the model I have to uniform the notations as well. The cooperating mobility management elements in my approach will be the Mobile Nodes (MNs) attaching to Mobility Access Points (MAPs) as a subset of the network of Mobility Agents (MAs).

2.1 Mobility Management Model - MMM

The function of mobility management is to secure that the mobile device is continuously able to access the network services, receive and initiate calls or data traffic during its movement. Location management is a sub component of mobility management with the aim to be able to determine the location of the mobile device with certain accuracy even when the mobile device is moving. There are a number of position management solutions in the literature for the different mobile network technologies. Generally all of them divide position management into two phases: position update and call delivery. Independent of the solution their usage generates certain cost in the network. This cost could include the amount of traffic transferred during signaling, resource and processing cost occurred in the network nodes, costs of using the radio interface or the extra steps needed to setup a secure connection.

If the mobile updates its current position more often the network can have more accurate location information hence the call delivery is simpler and faster. In this case the position update cost is higher, but the cost of call delivery is lower. With less frequent position updates, the costs are the other way around. Finding a good balance is a typical optimization problem.

The position management strategies operate with different efficiency in various network structures. Therefore it is important to compare the cost of different position management strategies during the planning of a concrete network to be able to choose the optimal solution. I have not found such a wide comparison model in the literature.

Thesis 1.1 I have created a Mobility Management Model (MMM), that is able to compare the main mobility management strategies in the literature from signaling, processing in the nodes, air interface usage and security point of view. I have defined cost functions for the different aspects [J7,J9,J11,C7,C8,C11,C14].

I investigated mobility management as an abstract problem regardless of actual technical solutions or serving network. I tried to grab the most significant properties of the mobility that is worth to discuss within the scope of the modern mobility protocols.

The mobility management model (MMM) is an extension and generalization of the simple mobility model in the literature [57]. My first step during its creation was the preparation of the network’s and the mobile’s entity model. One possible realization of the network modeling is
using a concrete, specific network representation to describe a protocol. The obvious disadvantage of the above is that it cannot be used for other protocols and in other contexts. Therefore in many Dissertations, papers the network is described by a simple, general parameter, like the mean/average distance between the nodes. Any network can be described this way, but the power of the model is rather weak if one would like to compare the behavior of different protocols on certain network structures. Summarizing the experiences I combined the two solutions, defined the general characteristics \((n)\), which are independent of the technology and also introduced a special, defined parameters needed to properly describe the protocol investigated \((c)\).

2.1.1 Basic notations and assumptions

I define Mobility Management System as an application running on network nodes that helps to locate the mobile equipment towards its unique identifier (the IP address for example).

- The specific network with all its parameters is denoted with \(N\).

- The Mobile Nodes \((MN,\) alias mobiles, moving entities, users) are the mobile equipments who want to communicate with any other mobile or fixed partner and move between the radio access points. The number of MNs in the model is denoted by \(n_{u}\).

- There are Mobility Access Points \((MAP,\) alias cells), these are the only entities that are capable to communicate with the mobile nodes via radio interface. All mobility access points have their own geographical area. While the MN moves in an area, it is always connected to the representative of the current area. The number of mobility access points in the model is denoted by \(n_{m}\). The user can connect to MAPs with handovers from the neighboring MAPs, each user is connected to only one MAP at a time. The neighbor MAPs could use even different access technology than the current MAP and they could be located the very same geographical place as well, the model does not requires one access technology in the whole network. The number of neighbor MAPs of MAP \(i\) is denoted by \(n_{i,m}\).

As I mentioned before MAP can mean a set of access points or even a network. This type of 'collapsed' MAP (Figure 1) is used when we do not want to focus on a part of the access network, or to include this part of the network into the cost calculation.

![Figure 1: The illustration of the 'collapsed' MAP.](image)

- The Mobility Agents \((MA)\) are network entities running the mobility management application. The number of MAs in the model is denoted by \(n_{a}\).
- There is a network (core network and access 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 are not part of the mobility management application and the edges can be various links (even radio links) for the data communication between the vertices. I denote these as Network Elements (NE).

With this definition one can see that most of the functional entities of the current mobility protocols and others under development can be generally described. (Details and proof of the above statement is given in Section 2.2.)

I simplify the problem of mobility management to a protocol that finds the correct, marked Mobility Access Point where a given Mobile Node is attached. To create this model I need some practical assumptions:

- A Mobility Access Point is always a Mobility Agent.
- All the nodes presented above are logical entities i.e. Mobility Access Points can mean a set of physical access points.
- Mobile equipment can communicate with multiple Access Points at the same time but one connection is necessary and enough to maintain the correct communication. The mobile can also attach and detach from any Mobility Access Points. At this point I assume that the mobile node is administrated only at one agent. This means that the problem of finding the mobile node is the same as finding the correct access point. (It is not difficult to enhance the discussion to the case where the MN can be found at multiple MAPs but it is out of the scope of the Dissertation.)
- The nodes in the network communicate and find each other using a given protocol or method (for example via IP routing). For this reason this part of the mobility protocols is not discussed.

The above definition and assumptions suit my aim that is to investigate the properties of various management strategy approaches, since the number of messages sent and the number of tasks should be completed can be calculated. With assigning appropriate cost parameters to each message or task one will be able to model exact mobility management solutions and can analyze them. I am going to give an example in Section 2.4.

### 2.1.2 Network Graph and Node Mobility Parameters

In this Section, I introduce how I will model the network structure on which the Mobility management systems work. To derive the main parameters I will have to model the behavior of the Mobile Nodes as well. There will be general and algorithm specific parameters derived. Secondly, I present the cost dimensions include to my model. These cost parameters related to the “signaling on the links” as a bandwidth and inter-working equipment usage (or even QoS and Service Cost ratio), the “processing in the nodes" which are taken into account only on the nodes running the mobility protocol, the “access cost" or “air interface usage" containing the cost of accessing the fixed network (MAP-MN attachment) or explicitly for example the battery consumption of the MN and the “security cost”, which is 'reversed' definition, the cost caused by cracking a message with a defined probability.
A few parts of parameters and functions is applied and reworked from an existing model and classification system of literature [57]. My mobility modeling system is more complex and works from different aspects of the problem. By introduction of elements published in [57] I will sign precisely which is not my work.

2.1.2.1 Modeling the network

As the first step I go through existing comparison works because in many papers, the network is modeled in order to emphasize the properties of a single protocol compared to another one. This approach is not flexible since new protocols cannot be imported to the comparison and also the little modifications in the protocols are difficult to follow but a model of this type sometimes essential for a protocol. Let us see the two main approaches of network models. One approach to describe the network is to give global parameters like a general average distance between nodes. It is true with this approach any kind of network could be described since the parameters can mostly be derived although the method to derive them is often not presented in the works. However, as we will see, introducing these parameters is not enough to compare most of the protocols because they cannot emphasize the benefits of each. (See [3], as an example.)

Other works model the network with given network structure so each protocol is examined in the environment it was designed into. Clearly, it is often essential to make appropriate examination but it makes difficult to extend the discussion. For example, when a GSM cell structure is used, no vertical handovers are taken into account. Another mobility protocol might have a different structure of covering the same geographical region when the graph, describing the network might not even be able to be drawn on a plane that will be the case in my model. (See [4], as an example.)

Summing up the requirements I introduce a method to derive global parameters from any kind of network to get the benefits of the first approach and I show a method how the protocol specific structure parameters can be derived. This generalizes the discussion while keeping some important specific characteristics.

2.1.2.2 Deriving parameters of a given network

Let me have a given network topology with a given MN behavior. The network is modeled with a graph and so are the possible movements of the mobile nodes. Thus the initial model is a weighted adjacency matrix for the network and a handover frequency (intensity) matrix for the Mobile Nodes movement.

Let me assume that the aggregated behavior of the Mobile Nodes can be modeled with a finite state continuous Markov chain (the handover or call arrival rate than is a Poisson process with various intensity parameters as in many works, e.g. [2]). The chain is given with a rate matrix \( B \). In this matrix, all the possible (in practice: the practically possible) MA-s are listed where the Mobility application runs. (These MAs can also denote single access points, bigger networks or the Home Agent if desired.) The number of MAs is \( n_a \) and so the matrix will be an \( n_a \times n_a \) matrix where each element \( b_{ij} \) denotes how frequent the movement of the mobile is from \( MAP_i \rightarrow MAP_j \). If an MA is not a MAP then there are 0 values in its row and column (i.e. I threat it the same way that the MN cannot or never attaches to it). From the rate matrix the transition matrix \( B_{\Pi} \) can be determined easily. I assume that the matrix \( B_{\Pi} \), without the non-MAP nodes, is practically irreducible and aperiodic that implies that the chain is stable and there exists a stationary distribution. This will be denoted by a density vector \( \pi \). In this
vector, the $i$-th element denotes the probability of the MN being located under the $i$-th MAP. This is the same probability as the relative number of handovers from and to the $i$-th MAP. (For MA nodes that does not support access point functionality, there is an element in the vector with 0 value.)

Let me have the corresponding network graph given with its weighted adjacency matrix: $A$. This matrix should include all the nodes in the network where the mobility application runs (all the MAs again) so has the same $n_a \times n_a$ size as matrix $B_Q$ and $B_\Pi$. The weights are relative values thus $w_{ij} = 1; w_{jk} = 2$ means that the relative cost (or any kind of measure) of any mobility management signaling from $j \rightarrow i$ is the double of the cost from $i \rightarrow j$. Several interpretations of the weights are possible for example relative delay or jitter or the number of routers on the path, etc. With the Floyd algorithm [81] the optimal distances between the nodes can be calculated (even with weighted or directed edges as well). The distance between nodes will be the sum of weights on the shortest (cheapest) path from one to the other. Let this result matrix be given by $A_d$. In the $i$-th row of the matrix, the distances from MAs are listed. Let the distances from the HA (Home Agent), - a special MA, - be given with the vector $a$. I will have parameter $w$ to denote the average of the weights in the network. It can be calculated by summing up the elements of $A$ and dividing it with $n_a^2$. Parameter $m$: Let parameter $m$ denote the average depth level, as in [57], that is the average sum on weighted edges on the shortest (cheapest) path from the MN to the HA. Clearly, the average number of vertices among the path is $m + 1$ if $w_{ij} = 1 \forall ij$. I will use matrix $A_d$ and vector $a$ to calculate this parameter. Both have to be normalized with the average weight of edges in the network ($w$). Now $mw$ can be calculated with determining the weighted average of the distances where the weights are the probabilities that the node is under a given MAP.

$$m = \frac{a \ast b}{w}, \quad (1)$$

where $\ast$ stand for the scalar product. One can see that the nodes which are not MAPs have a 0 multiplier and do not count in the average distance as expected. These parameter show the real average number of edges and distances along the shortest path from the HA. If there is a call or delivery request, I suppose that it is routed on this optimal path towards the current MA of the MN. In this sense these are the smallest values of my parameters. (After calculating $m$ one can further manipulate it for example multiply it with the probability of choosing the optimal path for each protocol if needed, etc.)

Parameter $g_T$: I will have another parameter like $m$ that is the average distance between two nodes who handles the MNs handovers (It is similar to $g$ in [57]). They might be connected, but they can also be quite far from each other logically due to different technologies especially in the case of vertical handovers. So as I see this parameter has to denote the weighted average value of the length between every two neighboring MAs where the mobile can attach. Then it is calculated as follows:

$$g_T = \frac{b \ast tr(A_d \cdot B_\Pi)}{w}, \quad (2)$$

where $tr$ is a trace of matrix, which means the diagonal elements in a vector of the matrix.
I denote it with $g_T$ since this parameter will have the most effect on the Tracking-like management solutions as we will see.

**Parameter $g_H$:** This parameter denotes how far is the nearest hierarchical junction to register in average if we consider the optimal covering tree of the network with the HA in the root. The junction node is the nearest common node of the paths from HA to the old and the new FA of the MN. In Figure 2 MN moves to position 2, in this case the nearest hierarchical node is MAP $B$, and $g_H$ represents the distance between MAP $A$ and MAP $B$. (In most cases, the optimal tree structure is not possible to achieve since the different service providers will not mesh their networks: approximate values can be used instead.) My hierarchical structure will be built up using two main parameters. First is the number of MA nodes: $n_a$. The other parameter is the average number of neighboring MAs that can be accessed via a wire from a given node: $w_{nw}$. It should be also weighted with the probability density of the MN:

$$w_{nw} = \frac{(b \cdot \text{sign}(B_{ii}) \cdot 1)}{n_a}. \quad (3)$$

**Parameter $g_C$:** This parameter will denote the average distance of MAPs from the main MA of a Location Area (LA) in the Cellular-like approaches. Together with this parameter we can compute the number of MAPs in a LA ($n_{LA}$) and the average probability of making handovers out from the cell ($p_{LA}$). These will be introduced later. It is an NP full problem to calculate the optimal cell and LA structure, but there are algorithms approaching it very good in some sense. For example [6] solves this problem under additional constraints and limits the maximal paging cost. In my numerical simulation I have run the algorithms developed and published in [5] and obtained $g_C$ with them. However, concerning the NP fullness two important notes have to be mentioned. When talking about an existing network, these parameters can be calculated easily. If I work with small network patterns, the optimal cell structure can be selected from the not too many options.

![Figure 2: Presentation of the network parameters.](image-url)
2.1.2.3 Modeling the mobile node

As we have seen, matrix $B_Q$ describes the movement behavior of the MN, handover-wise. If one sums up the $i_{th}$ row in this matrix it gets a rate how frequent the MN moves from the $i_{th}$ MA (MAP) with a Poisson-process. Let $\lambda$ denote the average parameter of the Poisson-process (at each MAP) and so denote the rate of handovers for a general MN anywhere in the network.

The other parameter that can be introduced in a similar manner is the rate of receiving a call: $\mu$. This parameter can also be time- or location-dependent. I take its average value as I did in the case of, and I assume it is constant in the examined very small time interval just like I did in the case of matrix $B_Q$ and through the whole modeling.

Using the achievements in, $\rho$ is the "mobility ratio" meaning the probability that the MN changes its FA before a call arrives. The good thing in my notation is that various movement modeling can be embedded into it with varying $B_Q$.

These parameters, $\lambda$, $\mu$, $\rho$ are introduced with the same determination as in [57].

2.1.3 Cost functions methodology and cost constants

The four main classes of cost types and the corresponding cost constants that can describe the technology level will be introduced here.

The main reason is that the same kind of protocol can be implemented on various network layers that influences its signaling and processing needs of each protocol message. Another reason to introduce such network topology and mobility strategy independent cost constants is that the underlying networking equipments might have very different characteristics so it might be important to test the behavior of the Mobility Management System on different serving networks to find the most suitable one. For these reasons, one will always have to tune these parameters according to the very implementation used. For my numerical example values see Section 2.3 where I also show that modifying the ratio of some (or some set of) parameters (for example the registration and packet forwarding cost) strongly affect the performance of a mobility management system.

2.1.3.1 Link related constants

First class of the cost constants are the link related constants (see. Table 1). They are introduced since one of the most important properties of a network is the bottleneck of uplink and downlink bandwidth especially when the service has to satisfy Quality of Service (QoS) requirements as well. As it was described, my network model does not necessarily show the real network topology. Each edge in the graph denotes the link from one MA to another. There might be several routers and subnetworks among the path. The parameters introduced here gives one unit signaling cost in each direction.

2.1.3.2 Node related constants

The node related constants model the cost of resources in the MA nodes (the vertices of my graph). For example the cost of creating a packet might be different for different protocols (number of headers, if there is tunneling, etc.) so these constants should be adopted to the examined protocol but also might be different using equipments of different vendors too.
Table 1. The cost constants

<table>
<thead>
<tr>
<th>Signaling cost constants</th>
<th>( c_u )</th>
<th>The unit cost of one update on a link.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( c_d )</td>
<td>The unit cost of one delivery on a link.</td>
</tr>
<tr>
<td>Processing cost constants</td>
<td>( c_r )</td>
<td>Registration cost, the cost of the process in the MAP when a MN node wants to attach.</td>
</tr>
<tr>
<td></td>
<td>( c_f )</td>
<td>Forwarding cost at a MA.</td>
</tr>
<tr>
<td></td>
<td>( c_m )</td>
<td>This is the constant cost of modifying some node related records in a MA.</td>
</tr>
<tr>
<td></td>
<td>( c_{ec} )</td>
<td>The cost of building up a message.</td>
</tr>
<tr>
<td></td>
<td>( c_{rc} )</td>
<td>The cost of recapsculating or rebuliding a message.</td>
</tr>
<tr>
<td></td>
<td>( c_{dc} )</td>
<td>The cost of decapsulate or open the message at an endpoint. In many cases ( c_{dc} = c_{dc} + c_{ec} ).</td>
</tr>
<tr>
<td>Air interface cost constants</td>
<td>( c_{au} )</td>
<td>The cost of uplink message between the MN and the MAP.</td>
</tr>
<tr>
<td></td>
<td>( c_{ad} )</td>
<td>The cost of downlink messaging between the MN and the MAP.</td>
</tr>
<tr>
<td>Security cost constants</td>
<td>( c_{wu} )</td>
<td>The amount of update data to be ensured on a link, in fact the payload. It could contain the security related processing cost also.</td>
</tr>
<tr>
<td></td>
<td>( c_{wdb} )</td>
<td>The amount of delivery data to be ensured on a link from HA to the hierarchical junction, in fact the payload. It could contain the security related processing cost also from HA to hierarchical MAP.</td>
</tr>
<tr>
<td></td>
<td>( c_{wdu} )</td>
<td>The amount of delivery data to be ensured on a link from hierarchical junction to the MN, in fact the payload. It could contain the security related processing cost also from hierarchical MAP to the MN.</td>
</tr>
</tbody>
</table>

2.1.3.3 Mobile equipment connection related constants

The access interface and connection related constants will denote of the unit cost of using the access interface. This access interface can denote radio access or the cost when one connects his laptop to an ethernet cable and there is load on the network from the laptop (MN) to the MAP (server of the network). I will collect the costs of this group of tasks in two parameters: upload and download. It is easy to see that the cost might be different for upload, download especially if download is a simple broadcasted paging (mainly MAP resources needed) while upload is a registration message that needs more resource (battery) at the MN side. As I mentioned in Section 2.1.2.1 the layer 2 handoff costs, for example the movement detection, Access Point (AP) searching, and AP reassociation can be taken into account here.

2.1.3.4 Security cost class

As I mentioned before in the introduction the security is key task in the mobile networks. The cost of security solution is also taken into count to compare the different mobility approaches. The signaling-, processing- and access cost are well defined and described, but the
cost of the security is an abstract definition. My assumption takes the amount of update- \((c_{ua})\) or delivery \((c_{ud})\) data to be ensured on a link, in fact this is the payload. From other point of view these are the loss when the message is cracked. These are summarized to the full path and weighted with the probability of cracking a message of the mobility protocol. This probability is determined by the following:

\[
P_{\text{crack}} = 1 - \frac{p_i^{\text{sec}} - p_{\min}^{\text{sec}}}{p_{\max}^{\text{sec}} - p_{\min}^{\text{sec}}},
\]

where \(p_i^{\text{sec}}\), the security strength of the actual mobility management solution, which is between \(p_{\min}^{\text{sec}}\) and \(p_{\max}^{\text{sec}}\). There are more security classification similar to this in the literature, which can be used in the cost calculations. In my numerical simulations and examples the methodology one of these [58][59] is used.

### 2.1.3.5 Example of cost diversification

Here I try to underpin the above classification. I want to give an example but of course many kinds of setups are possible. When a MN moves (leaves its old MAP and connects to a new one) different events happens in layer 2 and layer 3. The costs of these events, the layer 2 handoff, the agent discovery and the actual registration are incorporated to interface and air interface cost in my Dissertation. The first one depends on the access technology and can be accomplished in different ways. For example in the case of IEEE 802.11b WLAN it means the AP changes, and can be split into three parts: movement detection, AP searching, and AP re-association. In order to handle this difference of layer 2 handoff of access technology I can take into account this cost using an average layer 2 handoff cost. Agent discovery is another process in layer 3, during this phase, the Mobility Access Point advertises their services on the network by using a Discovery Protocol for example ICMP Router Discovery Protocol (IRDP) in Mobil IP. The Mobile Node listens to these advertisements to determine where it is connected to, or if it is connected to its home network or foreign network. The Discovery Protocol advertisements can carry the types of services MAP will provide such as reverse tunneling and Generic Routing Encapsulation (GRE) and the allowed registration lifetime or roaming period for visiting Mobile Nodes. Rather than waiting for agent advertisements, a Mobile Node can send out an agent solicitation. This solicitation forces any agents on the link to immediately send an agent advertisement. When the Mobile Node hears an Agent advertisement and detects that it has moved outside of its last network then registers its current location during registration process.

### 2.2 The main management strategy classes

In this Section, the four main classes of mobility management protocols introduced in Section 2 are shortly described and modeled with their signaling-, processing-, air interface-, and security cost functions. Some of the signaling cost functions are the same as in [57], these will be signed exactly.

The cost function as the model for the mobility approach can be any kind of utility function depending on the purpose of use. It is constructed as a calculation of expected utility as a function of network topology descriptors and cost constants with respect to mobility i.e. the conditioned probability of changing the attachment point with \(\lambda(t)\) intensity on not being paged with \(\mu(t)\) intensity generally:
\[ C = E\left[ \int_{-\infty}^{\infty} f_h(n,c)dP(\lambda(t)) + \int_{-\infty}^{\infty} f_p(n,c)dP(\mu(t)) \right], \quad (6) \]

where \( \lambda, \mu \) are the intensities of a Poisson process and \( n \in \{ \text{“network parameters”} \}, c \in \{ \text{“technology constants”} \} \). Functions \( f_h, f_p \) are unique for each mobility management strategy and mostly are a linear function of the network parameters and the technology constants. They tell us the cost of one handover or one paging for each protocol respectively. Since the integral is taken with respect to a Poisson process it will be a simple sum and with introducing the handover process conditioned on no paging the above equation simplifies to:

\[ C = \rho \cdot f_h + (1 - \rho) \cdot f_p, \quad (7) \]

where \( \rho = \frac{\lambda}{\lambda + \mu} \).

Although the Poisson process is not able to model the packet switch, sophisticated characteristic traffic in mobile networks, but it is capable to model number of incoming packets per time unit. There are more papers with Poisson model in the literature [82][83][84][84].

### 2.2.1 Centralized approaches

This group of managements contains various protocols like Mobile IP [10]. The common in these management structures is that the MN always sends location update messages in case of handover to a central or a central group of management nodes, that maintains a database and has up-to-date information about the exact location of the MNs (Figure 3).

The Central Agent (or Central Agents) always has to know the exact location of the Mobile Node in order to inform the correspondent nodes, or to forward the requests to MN using tunnel or source routing. The followed method by Central Agent depends on the mobility solutions. For example in Mobil IP protocol the Home Agent as a Central Agent, forwards the packets using tunnel, but in SIP based mobility the Registrar server answers the IP address of MN to the SIP Proxy, which controls the communication.
The cost functions of Centralized Approach:

\[ C_{\text{SIGN}}^{\text{cent}} = \rho mc_c + (1 - \rho) mc_d \]  
\[ C_{\text{PROC}}^{\text{cent}} = \rho(c_c + (m - 1)c_f + c_m) + (1 - \rho)(c_{ec} + (m - 1)c_f + c_{inf}) \]  
\[ C_{\text{AIR}}^{\text{cent}} = \rho c_{au} + (1 - \rho)c_{ad} \]  
\[ C_{\text{SEC}}^{\text{cent}} = P_{\text{crack}} \cdot (\rho mc_{su} + (1 - \rho)mc_{sd}) \]  

As we have just seen above the cost functions are obvious and simple. In the processing function there is only one encapsulation, one decapsulation and \( m-1 \) forwarding cost, which do not need a big computation capacity. The application of such a protocol has a second main advantage along with its simplicity, namely these approaches can be built by installing a Central Agent in the network and by running an IP-level software module on the MN. There is no need to change any other entity in the network, therefore it is cheap and easily installable. But we have to take a relevant disadvantage into account, centralized mobility puts extraordinary high overload on the bearer network and uses non optimal routing. However, this solution is far from the optimal, still the most of the mobility implementations use the same kind of this centralized approach. Transport layer mobility, for example mSCTP (multi-homing Stream Control Transmission Protocol), HIP and Application layer mobility for example SIP, as it was mentioned above, belong to this centralized approach. (The mSCTP allows an association between two end points to span multiple IP addresses or network interface cards. It supports to keep alive the TCP sessions during IP address change. HIP separate the node identification and location information in the network layer. This solution affords the applications a permanent network layer. SIP provide the possibility of many Voice over IP services.)

Despite these advantages all of these protocols need a centralized management to accomplish the mobility, which is far from the efficiency it could have.

2.2.2 Hierarchical solutions

Instead of the global management node regional management system can be used to reduce the signaling traffic by maintaining the location information locally. For this reason we can use the MAPs and MAs as local agents, that have database to store the actual IP addresses of MN. So we can consider this hierarchical network structure as a tree of MAP, MA and other network node with Central Agent in the root of the tree. The main idea is that the update information is sent only to the nearest MA on the network, if the MN moves within the subnetwork managed by this entity. Parameter \( g_{H} \) denotes this distance as it is explained in Section 2.1.2.2

Let us now take a look at the network tree of the hierarchical mobility. One path from the HA leads to the old MA of the MN and another one leads to the new one. It is enough to send the location update message to the nearest common router with the old path and the new path, as one can see on the Figure 4.
However, typically there are other nodes placed between the MAs. Now we take the best case for the signaling optimization and consider every node in the tree as a MA, or a set of MAs and MAPs. Within the subnetwork controlled by one MA a subnetwork IP is assigned to the node, and changes when the node changes its point of attachment in this level of the tree. In this approach Central Agent knows the IP address of the MA, under that in the network tree the MN is located, or even moves. The costs function changes compared to the centralized solution, because the location information is sent only to the nearest MA ($g_H$ distance from the MN). The cost functions:

\[
\begin{align*}
C_{\text{SIGN}}^{\text{hier}} &= \rho g_H c_u + (1 - \rho) mc_d \\
C_{\text{PROC}}^{\text{hier}} &= \rho (c_r + (g_H - 1)c_f + c_m) + (1 - \rho)(c_{ec} + (m - g_H - 1)c_f + c_{rc} + (g_H - 1)c_f + c_{dc}) \\
C_{\text{AIR}}^{hier} &= \rho c_{au} + (1 - \rho)c_{ad} \\
C_{\text{SEC}}^{\text{hier}} &= P_{\text{crack}}'(\rho g_H c_{su} + (1 - \rho)((m - g_H)c_{adh} + g_H c_{adh})
\end{align*}
\]

The advantage of this method is the more optimal functionality, and less load on the bearer network. However, the change of some other entity is needed in the network, therefore the solution is more expensive. An example for such solution is the Hierarchical Mobile IP (HMIP) [1].

### 2.2.3 Tracking-like solutions

In the tracking-like approaches each mobile node has an entry in a Central Agent like in other solutions. This CA stores the address where it received location update message from. It is the address of an MAP, and a next-hop towards the mobile node. The mobile node is either still connected to that MAP, or that MAP knows another next-hop MAP towards the mobile. Finally the mobile node can be found at the end of a chain of MAPs.

As it was introduced the main idea behind the tracking-like algorithm is that if the MN changes its point of attachment then it could be a good solution to send an update to the old MAP. The MAP that the mobile node moves away is called old MAP, the one it moves to is called new MAP. After this handover, called tracking handover the old MAP is able to forward the request towards the MN via the new MAP.
There is another kind of handover in tracking-like solutions, the normal handover. Normal handover occurs when mobile equipment updated its entry in the Central Agent by sending the address of the new MAP node to it.

The normal handover is similar to centralized solutions; it generates a lot of signaling traffic, but a tracking handover puts less, or no signaling to the network (see Section 2.2.3.1 and 2.2.3.2). But if an incoming packet arrives to the mobile node, I have to find it in a hop-by-hop manner, and send a location update message to the Central Agent, which are expensive. In tracking scheme a normal handover can be followed by some tracking handovers before another normal handover takes place. If the mobile node does not receive a packet between two normal handovers then less signaling is used, if a packet is received after some tracking handovers, more signaling is used compared to a centralized mobility scheme.

Thus the most important decision of tracking-like mobility is when to make a normal handover, and when to make a tracking handover. In the tracking-like approaches each mobile node has an entry in a Central Agent like in other solutions. This CA stores the address where it received location update message from. It is the address of an MAP, and is a next-hop towards the mobile node. The mobile node is either still connected to that MAP, or that MAP knows another next hop MAP towards the mobile. Finally the mobile node can be found at the end of a chain of MAPs. One can read more about these protocols in works: [8], [10], [9].

I distinguish between two different tracking-like solutions based on the kind of tracking handover: wireless tracking and wired tracking. In case of tracking handover of wireless tracking the mobile sends the address of the new MAP node to the old MAP node over the air interface (for reference see LTRACK [9]) while in case of the wired tracing the information is sent over the wired network [10].

The optimal number of tracking handovers between two normal handovers has to be calculated in both cases. I use the achievements in [9] to determine these parameters and define the optimal cost function.

### 2.2.3.1 Wireless tracking

In case of tracking handover of wireless tracking the mobile sends the address of the new MAP node to the old MAP node over the air interface (Figure 5).
If the mobile node can communicate only to one MAP (hard handover) then the address of the new MAP has to be sent to the old MAP just before the handover takes place but if the mobile is capable of communicating to more than one MAP simultaneously (soft handover) then the address can be sent any time during the handover. If the mobile suddenly loses the connection to the actual MAP then after finding new MAP it makes a normal handover in order to establish path to itself. The cost functions:

\[
C_{\text{SIGN}} = \rho P_H g_H c_u + (1 - \rho)(m c_d + M[h_r] g_T c_d + (1 - P_0) g_H c_u)
\]

\[
C_{\text{PROC}} = \rho((1 - P_H)(c_r + c_m) + P_H(c_r + (g_H - 1)c_f + c_m)) + (1 - \rho)(c_{ec} + (m - 1)c_f +
\]

\[
+ P_0 c_{dc} + (1 - P_0) (M[h_r]((g_T - 1)c_f + c_{rc}) + c_{dc} + (g_H - 1)c_f + c_m))
\]

\[
C_{\text{AIR}} = \rho((1 - P_H) 2c_{au} + P_H c_{au}) + (1 - \rho)c_{ad}
\]

\[
C_{\text{SEC}} = P_{\text{crack}} \cdot (\rho P_H g_H c_{au} + (1 - \rho)((m - g_H) c_{edh} + (g_H + M[h_r] g_T) c_{edh} + (1 - P_0) g_H c_{au})
\]

where \(M[h_f] [9]\) is the optimal number of tracking handovers, and \(P_0\) is the probability that there was no tracking handover.

The tracking handover in this case does not put any signaling load on the network except the load of the air (access) interface. For the cost functions see Table 2. The mobility management named Location Tracking (LTRACK) [9], introduced in 2003, belongs to this type of mobility management.

### 2.2.3.2 Wired tracking

Wired tracking differs from wireless one in the method of the tracking handover. In this case the mobile sends the address of the new MAP node to the old MAP node through the wired network as it can be seen (Figure 6). This handover puts some signaling load on the network, but it is not significant, and less than a signaling load in case of normal handover. The advantage of this method is that it saves the air interface resources.

![Figure 6: Basic operation of Wired Tracking Approach.](image)

Partly similar cost functions to the wireless tracking can be derived:
The well-known Handoff-Aware Wireless Access Internet Infrastructure (HAWAII) [10] is classifiable to the wired tracking algorithms.

As we have seen when a normal handover occurs in tracking solutions, there is a signaling message sent to the Central Agent. It is obvious that a hierarchical mobility layer structure could be used in the way that it is used in every handover of the Hierarchical approaches. This is the reason why I used $c_H$ in the cost functions of tracking solutions. This makes the signaling cost of this solution lower than or equal to the cost of Hierarchical protocols. The processing cost could be less optimal because of the several decapsulation cost of the packet. To compute the number of optimal tracking handovers $H$ between normal handovers I adopt achievements from [9]. For the basic model the cost functions of wireless tracking can be treated as a continuous one and can be derived. This provides a fast and easy solution for computing the optimal value of $H$ with taking looping into account as well. If I extend my model with the effect of loop removal, it is clear that the cost function for tracking approaches remains the same, but values of the state probabilities ($P_H, P_0$), and the expected point of return ($M/hr$) [9]) will be different.

\[ C_{\text{wired}^{\text{SIGN}}} = \rho((1-P_H)g_T + P_H g_H)c_u + (1-\rho)(mc_d + M[h_r]g_Tc_d + (1-P_0)g_Hc_u) \]
\[ C_{\text{wired}^{\text{PROC}}} = \rho(c_r + (g_T-1)c_f + c_m) + (1-\rho)(c_{ec} + (m-1)c_f + P_0c_{dc} + \\
+ (1-P_0)M[h_r]((g_T-1)c_f + c_m) + c_{dc} + (g_H-1)c_f + c_m)) \]
\[ C_{\text{wired}^{4R}} = \rho c_{au} + (1-\rho)c_{ad} \]
\[ C_{\text{wired}^{\text{SEC}}} = P_{\text{crack}} \cdot \rho ((1-P_H)g_H + (1-P_H)g_T)c_{au} + (1-\rho)((m-g_H)c_{adh} + \\
+ (g_H + M[h_r]g_T)c_{sdc} + (1-P_0)g_Hc_{au}) \]

(11)

2.2.4 Cellular-like solutions

To mobility problem there are cellular-like solutions as well. The idea behind this kind of approach comes from the GSM protocol. Basically this kind of protocols can be used in lower level of the hierarchical network. In a cellular systems the nodes do not know the topology of the network and the exact location of the MN. Packets to MN are routed by hop-by-hop manner, it is little the same as in tracking approaches. However, in most cellular solutions the routing is generally accomplished in layer 2, so the MAPs only have to know on which of its outgoing ports to forward packets. The solution builds strongly on the fact, that from the large number of mobile nodes only a small percentage are receiving data packets. For this reason I can define optimized -, well-defined areas, called paging area or location area, and it is enough to know in which paging the idle mobiles are moving.

The advantage of this approach are the quick handover mechanism in lower layer and cheap passive connectivity as it can be seen through the cost functions on table 2 as well. The disadvantage is that the building of the network has to be careful and too many paging will cause an extreme increase in the costs. (One can see how the cost decreases with the growth of the mobility ratio $\rho$ in Section 2.3.)

Two constants, related to the network topology are very important: - $n_c$: The average number of MAPs in a paging area; - $n_d$: The number of paging areas in the whole network.
2.2.4.1 Standard Cellular-like solutions

One well-known example for this approach is Cellular IP (CIP). In this case the hop-by-hop manner routing leads the packet only to the domain border of the paging area. From this point of the network to the mobile the nodes in the paging area do not store any information about the idle mobiles, accordingly in case of packet addressed to an idle mobile the paging area is flooded with the packet by broadcast message. I have a general model for it, see Figure 7.a.

When an idle mobile realizes that the network searches for it then it changes its state to active, sends normal update to inform the paging area about its exact location, and receives the packets. After the normal update the broadcast sending becomes unnecessary. It is very important to perform this procedure quickly in order to the bearer network and the air interface could be spared the high cost of broadcast messages.

The cost functions of this solution are the following:

\[
C_{\text{SIGN}}^{\text{cell}} = \rho (1 - P_c)g_{HC}c_{ad} + (1 - \rho)((m - g_{HC}) + (n_c g_{HC})c_d + g_c c_d) \\
C_{\text{PROC}}^{\text{cell}} = \rho ((1 - P_c)(c_e + g_{HC}c_f + c_m)) + (1 - \rho)(c_{ec} + (m - g_{HC} - 1)c_f + c_{rc} + \\
+ (g_{HC} - 1)n_c c_f + n_c c_{dc}) \\
C_{\text{sec}}^{\text{cell}} = \rho ((1 - P_c)c_{au}) + (1 - \rho)(n_c c_{af} + c_{au}) \\
C_{\text{AIR}}^{\text{cell}} = P_{\text{crack}} \cdot (\rho (1 - P_c)g_{HC}c_{su} + (1 - \rho)((m - g_{HC})c_{sh} + (n_c g_{HC} c_{sh}) + g_c c_{su})
\]

where \( P_c \) is in case of Cellular-like approaches the probability of entering to a new paging area.

2.2.4.2 Manet-like solutions

I will present another micro-mobility protocol based on the MANET (Mobile Ad Hoc Network) [11] that uses the technique of wireless ad-hoc networks (Figure 7.b).

The "MANET [11] in the paging areas" solutions introduced could be the best solution when we would like to save the infrastructure cost and the air interface using is cheaper. In this management system it is assumed that all MN could be reached via other MNs. Paging areas are defined like in other cell-like solutions, but only one MAP exists in one page, through this

![Figure 7.a: Basic operation of Standard Cellular-like Approach, b: basic operation of MANET-like Approach.](image-url)
the packets are routed using an optimal MANET algorithm. Advantage of this solution also is that signaling cost can be saved with correct MANET protocol in a page. However, in the suboptimal case some mobiles could not be reached, and aggregate air interface cost can be high.

\[ C_{\text{manet}}^{\text{SIGN}} = \rho (1 - P_e) g \text{du} c_u + (1 - \rho) (m - g_c + 1) + (P_M n_c (g_c - 1) c_d) + g_c c_u \]

\[ C_{\text{manet}}^{\text{PROC}} = \rho ((1 - P_e)(c_r + g \text{du} c_f + c_m)) + (1 - \rho)(c_{ec} + (m - g_c)c_f + + c_{rc} + P_M n_c (g_c - 2)c_f + c_{dc}) \]

\[ C_{\text{manet}}^{\text{AIR}} = \rho ((1 - P_e)(g_c - 1)c_{au}) + (1 - \rho)(P_M n_c g_c c_{ad} + c_{au}) \]

\[ C_{\text{manet}}^{\text{SEC}} = P_{\text{crack}} \cdot (\rho (1 - P_e) g \text{du} c_{su} + (1 - \rho)((m - g_c + 1)c_{sdh} + (P_M n_c (g_c - 1)c_{sdu}) + g_c c_{su}) \]

\[ (13) \]

### 2.2.4.3 Hierarchical Paging-like solutions

The main idea behind the Hierarchical Paging [12] is that not only the lower layer network is flooded with the packet but broadcast message is used to find the paging controller MA in the higher layer as well. With this functionality signaling cost could be saved because update messages are not sent to HA, but only to the MA which controls the page. But in case of calling the multilevel flooding causes high network load.

\[ C_{\text{page}}^{\text{SIGN}} = \rho (1 - P_e) g_c c_u + (1 - \rho)((m - g_c)n_d + (n_c g_c c_d) + g_c c_u \]

\[ C_{\text{page}}^{\text{PROC}} = \rho ((1 - P_e)(c_r + g_c c_f + c_m)) + (1 - \rho)(c_{ec} + (m - g_c - 1)n_c c_f + + c_{rc} + n_c (g_c - 1)c_f + c_{dc} + (g_c - 1)c_f + c_{m}) \]

\[ C_{\text{page}}^{\text{AIR}} = \rho ((1 - P_e)c_{au}) + (1 - \rho)(n_c c_{ad} + c_{au}) \]

\[ C_{\text{page}}^{\text{SEC}} = P_{\text{crack}} \cdot (\rho (1 - P_e) g_c c_{su} + (1 - \rho)((m - g_c)n_d c_{sdh} + (n_c g_c c_{sdu}) + g_c c_{su}) \]

\[ (14) \]
2.3 Compare different approaches

In this section I will present some general results on the models have made and as a second part I try to answer some interesting questions for example: What if a cellular cell system was implemented on layer 3 level rather than layer 2? What if MIPv6 kind of messages and low level solutions were used for a Tracking-like approach? I will also show how my model can be used to make vertical handover decisions.

I do attempt to give a few important examples for the type of investigations, that could be performed using my model. These investigation examples are beneficial for network planning, fine tuning and optimizing. I do not try to give an exhausting numerical analysis with my method here since this part of the Dissertation focuses on the modeling framework validation itself. For this reason I aim to verify the correct behavior of the examples in this section.

2.3.1 General differences between management approaches

Thesis 1.2 The MMM is able to choose the optimal mobility strategy or to compare existing management protocols for an \(N<\mathbf{A}, \mathbf{B}_\Pi>\) network and \(\varphi\) cost constant vector \([J7,J9,J11,C7,C8,C11,C14]\).

There are two approaches for using the MMM framework. The first one is if the network description \((\mathbf{u})\), the supposed/hypothetical cost constant vector \((\varphi)\) is available and the mobility management strategy with the lowest cost is determined. The other field of application is for comparing concrete, existing management protocols on one or more specific network configurations. In this case the strategy if the protocol is given, the cost constants specify the exact management method.

The description of the network is equivalent of defining matrices \(\mathbf{A}\) és \(\mathbf{B}_\Pi\). Afterwards the previously introduced parameters (Table 1) can be determined. The mobile entity’s handover frequency between two MAPs can be modeled by a Poisson process. Let \(\mathbf{B}_Q\) denote the rate matrix of the continuous Poisson chain describing the mobile’s characteristics. Transition matrix \(\mathbf{B}_\Pi\) can be deducted/reduced from \(\mathbf{B}_Q\). The network topology, connections between MAs are represented in matrix \(\mathbf{A}\). I compared the cost functions of the five main strategies in different aspects (Figures 8 and 9) using Mathematica [8] application.

In Figures 8.a one can see the difference between the approaches considering all the cost types (signaling, processing, air, security). It is clear that with the bigger frequency of handovers \((\varphi)\) the cost is bigger for the centralized-like, hierarchical-like and wired tracking-like approaches since each handover gives more signaling on the network. In the wireless tracking-like case if the number of handovers rise between the incoming calls it starts to save the costs of the rerouting of the packets. In the centralized-like ones, it is clear that the rarer there is an incoming call the less load the network has. The cost is obviously high in these case. The same case is printed on both figures, but the values of \(g_T, g_C\) network parameters are significantly less than \(g_H\) (more meshed network) on the right side figure. One can see that the wired tracking-like solution is getting cheaper as well and begins to behave as its tracking-like pair.
Figure 8.a: One can see the summed cost functions of centralized-like (one-dot-dash), hierarchical-like (two-dot-dash), wired (dashed) and wireless (solid) tracking-like, cellular-like (dotted) approaches here with the vary of the mobility ratio: $\rho$. The two figures show the costs on different networks.

The mobility ratio is fixed (a: $\rho = 0.3$; b: $\rho = 0.9$) in Figure 8.b. One can see that with bigger $m$ the cost is higher for the centralized and hierarchical solutions, the other approaches are independent from parameter $m$. As in previous case it is clear that the rarer there is an incoming call the less load the network has in case of cellular-like solutions.

Figure 8.b: The cost with the vary of $m$, with the same notation at $\rho = 0.3$ (left hand side) and $\rho = 0.9$ (right hand side).

On Fig. 9 the mobility ratio is fixed again to (a: $\rho = 0.7$; b: $\rho = 0.9$) respectively. On the other hand the cost of a single upload ($c_u$) to a single download ($c_d$) is exponentially changing from the half to the twice on the horizontal axis. Most of the solutions are more expensive if the upload is higher but it can be seen that the wireless tracking cuts this cost as expected.
Figure 9: The signaling cost constants vary dependency with the same notation at \( \rho = 0.7 \) (left hand side) and \( \rho = 0.9 \) (right hand side).

Figure 10 is a 3D plot, it shows the common affect of \( \rho \) and cost constants vary dependency. In this case the wireless approach is chosen.

Figure 10: The cost of wireless approach with vary of \( \rho \) and signaling constants.

I analyzed the full cost of approaches in case of example network environments. Three different, typical network topology are investigated with different handover intensities. The tree typical network scenario (Figure 11):

- **More meshed** (for example an LTE network on Figure 11.a.) : \( m=2, g_H=2, g_T=1, g_C=1 \), it is almost full meshed network.

- **Long-Tree topology** (for example a IMS network on Figure 11.c.) : \( m=9, g_H=7, g_T=7, g_C=7 \)

- **Wide-Tree topology** (for example an UMTS network on Figure 11.b.): \( m=3, g_H=2.5, g_T=7, g_C=2.5 \)
Figure 11: a: More meshed network example, a LTE network [62][78], b: Wide-tree topology example, an UMTS network [79], c: Long-tree topology example, an IMS network [80].

Table 2 shows the results.

<table>
<thead>
<tr>
<th>full_cost</th>
<th>More meshed</th>
<th>Long-Tree topology</th>
<th>Wide-Tree topology</th>
<th>More meshed</th>
<th>Long-Tree topology</th>
<th>Wide-Tree topology</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\rho = 0.3$ (lower freq. of handover)</td>
<td>$4$</td>
<td>$16.32$</td>
<td>$6.72$</td>
<td>$8.02$</td>
<td>$24.96$</td>
<td>$11.76$</td>
</tr>
<tr>
<td>$\rho = 0.9$ (high freq. of handovers)</td>
<td>$3.86$</td>
<td>$14.94$</td>
<td>$6.36$</td>
<td>$7.60$</td>
<td>$20.82$</td>
<td>$10.72$</td>
</tr>
<tr>
<td>CENT</td>
<td>$3.47$</td>
<td>$10.81$</td>
<td>$9.38$</td>
<td>$7.41$</td>
<td>$11.7$</td>
<td>$10.85$</td>
</tr>
<tr>
<td>HIER</td>
<td>$3.93$</td>
<td>$13.75$</td>
<td>$22.64$</td>
<td>$7.61$</td>
<td>$20.65$</td>
<td>$16.64$</td>
</tr>
<tr>
<td>WLESST</td>
<td>$7.04$</td>
<td>$103.2$</td>
<td>$18.48$</td>
<td>$4.54$</td>
<td>$19.8$</td>
<td>$6.54$</td>
</tr>
</tbody>
</table>

In more meshed network the performance of the approaches are nearly the same. Significant differences are noticed in the tree topology. The wireless tracking strategy performs well in more cases, but it depends especially on the air interface costs. Other obvious conclusion, which are mentioned before, for example: the centralized approach with high handover frequencies seems to be a bad decision in most of the cases, or the good performance of cellular like strategies in these similar cases.
2.3.2 Technology tendency via cost constants and simulation

To answer questions like 'What if a strategy like a cellular approach was implemented on a different technology level', I should give clear values of the cost constants. To do this I build up a simulation environment. OMNet++ [52] was used, which is a public-source, component-based, simulation environment with strong GUI support and an embeddable kernel. I have realized four IP layer mobility management, Mobile IPv4, Mobile IPv6 and LTRACK [9] on IPv4 and IPv6. I expect different technology constants for the two types of Internet Protocol however it is also unavoidable that two different kind of realized protocols differ in the constant values. During the simulation develop I followed strictly the protocols description and the result is presented in Table 3.

<table>
<thead>
<tr>
<th>Cost type</th>
<th>MIPv4</th>
<th>MIPv6</th>
<th>LTRACKv4</th>
<th>LTRACKv6</th>
</tr>
</thead>
<tbody>
<tr>
<td>$c_a$</td>
<td>499</td>
<td>777</td>
<td>481</td>
<td>695</td>
</tr>
<tr>
<td>$c_d$</td>
<td>1825</td>
<td>1821</td>
<td>1814</td>
<td>1843</td>
</tr>
<tr>
<td>$c_r$</td>
<td>187</td>
<td>1068</td>
<td>711</td>
<td>821</td>
</tr>
<tr>
<td>$c_f$</td>
<td>652</td>
<td>685</td>
<td>636</td>
<td>690</td>
</tr>
<tr>
<td>$c_m$</td>
<td>543</td>
<td>1239</td>
<td>538</td>
<td>541</td>
</tr>
<tr>
<td>$c_{cc}$</td>
<td>240</td>
<td>234</td>
<td>237</td>
<td>239</td>
</tr>
<tr>
<td>$c_{cr}$</td>
<td>0</td>
<td>0</td>
<td>235</td>
<td>238</td>
</tr>
<tr>
<td>$c_{dc}$</td>
<td>240</td>
<td>234</td>
<td>240</td>
<td>240</td>
</tr>
<tr>
<td>$c_{cm}$</td>
<td>136</td>
<td>181</td>
<td>138</td>
<td>165</td>
</tr>
<tr>
<td>$c_{cd}$</td>
<td>390</td>
<td>396</td>
<td>368</td>
<td>397</td>
</tr>
</tbody>
</table>

Shortly discussing the results I would like to point out an example. Taking a look at Table 3 it can be seen that in the case of MIPv6 the encapsulation cost ($c_{cc}$) is lower because there is no 'triangle communication' instead it uses the mechanism of constantly updated 'binding caches' while the registration cost ($c_r$) is higher because the IPv6 protocol overhead itself is higher. These facts intuitively verify my simulations. To prove the applicability of the simulation to my model framework I ran it on different network topologies and mobile movement setups and derived the expected value of each. With long run simulations the distribution concentrated near to the mean with low variation what proves the fact that these technology related constants are network and algorithm independent. The network dependent cost co for example the bandwidth measures are given in the weighted adjacency matrix $A$.

![Figure 12](image_url)

Figure 12: The left hand side Figure represents the cost of protocols in the ratio of 'accessing' and 'signaling' costs while the right hide side Figure does the same in the ratio of 'processing' and 'signaling' again (centralized-like (one-dot-dash), hierarchical-like (two-dot-dash) and wireless tracking-like (solid)).
In addition to the table above one is supposed to weight each type ('processing', 'signaling', 'accessing') of cost with different values since the cost of bandwidth is not in the same measure as the cost of processor load for example. In Figure 12 it can be followed for each how the cost of the mobility changes with the ratio of 'accessing' and 'signaling' then the ratio of 'processing' and 'signaling' cost respectively. If I know that in the future the mobile equipments and accessing technologies will become relatively cheap to the signaling resources then Figure 12 tells us that the *wired* protocol saves relatively the most with this tendency (of course on the given network and with the given implementation with the given mobility parameters, etc.).

### 2.3.3 How to use the model for vertical handover decisions

I will present how to use my mobility management framework to make vertical handover decisions. Consider a vertical handover system of two protocols on the same region. The weights on network edges will denote the relative cost of using the given protocol (cost constants) divided by the QoS it provides. As an example, let network 'A' be a locally maintained university WLAN network that can be used almost for free but provides bad QoS and network 'B' be a LTE network covering the same area which provides better QoS but expensive. The ratio of the relative cost in money and some QoS measure depends on the user (or installed service). The network matrix then splits to two parts. One is for network 'A' and the link weights (that basically shows relations between all the links) are multiplied by the above ratio expressing the user (anti)preference. If the value ratio value is higher then the user is less willing to use the network since it has the more basic cost.

To grab the meaning of the vertical handover decision I have to adapt the MN movement modeling part, the handover frequency matrix. If the network 'A' is too expensive, then the total handovers from the network 'B' are lower while backward handovers became higher expressing that the user is more willing to stay at the preferred part. Of course topologically the mobile nodes have to move within a given network too if they are making handovers.

In Figure 13 I depict an example how the vertical handover decision can be modeled within my framework. The two networks can be found on the top and the bottom. The nodes can move within the access points (MAP) of a network as a part of the normal operation. Note that the two Access network do not need to cover the same geographical area instead where vertical handover is possible I put a link into the Handover Frequency graph with $\eta$ and $1/\eta$ for each direction.

As one can realize I used a stochastic method to make vertical handover decisions. It is in the scope of my research to compare it to others but for now I try to emphasize why a stochastic method should be applied through an example. The total cost of QoS in access network 'A' is higher than in 'B'. On the other hand if the user mobility is higher than while it receives less traffic it is much more worth to stay at network 'A'. Since the mobility is a Poisson process, and the threshold depends on it (not just on its mean), it will vary according to a random variable too.
I used Mathematica for the implementation of the cost functions and the modeling and its capability for symbolic computations helps me to easily manipulate those rows and columns in the handover frequency matrix which describe the above motions and relations.

In Figure 14 one can see how the final cost of mobility changes with the vertical handover willing rate ($\eta$) from network 'A' to network 'B'. The point where $\eta = 1$ shows the cost for the situation when it is irrelevant for the MN which network it uses. One can see that it is not (necessary) the point with the optimal mobility cost $\eta^*$. In this current example the mobility cost is interpreted as a utility function for the user. The user is willing to optimize its cost and QoS ratio so will always set $\eta = \eta^*$ when making a handover and chooses network 'A' with probability $P[\text{Choose Network 'A'.}] = (\eta \tau) \exp(\eta \tau)$, since $\eta$ is the intensity of the Poisson process of making a handover.

The interpretation is the following. Let event $E1$, $E2$ be 'Vertical handover to Network 'A' before making a handover.', 'Not receiving a call.' respectively thus $P[E1|E2] = \frac{\eta}{\eta + \sum \omega \cdot w_i}$ where $w_i$ is the handover intensity to a given direction. Another interpretation can be done with the events $E3$ and $E4$ meaning 'Handover to Network 'A' before receiving the next call.' and
'Make no handover within Network „B”.' respectively thus \( P[E3|E4] = \frac{\eta}{\eta + \mu} \). The latest interpretations are important for us to understand the parameter \( \eta \). It effectively tells us the intensity of moving from network 'A' to network 'B'. If there was such a handover, then the backward handover is expected, with intensity of \( \frac{1}{\eta} \).

Of course as the quality in a network falls the user is more likely to switch to the other. In Figure 14 I depict the change of Utility in the function of the ratio while in Figure 15 I depict them for Mobile IP and Hierarchical Mobile IP according to both two parameters. In most of the QoS/Cost parameter cases one might find the optimal \( \eta^* \) rate going to infinity which means that it is worth to switch network immediately while it is not worth to switch back.

![Figure 15: On the left hand side figure one can see the 1/Utility (Cost) variation for the user using MIP protocol while on the right hand side for HMIP protocol. Both are depicted as a function of \( \eta \), handover probability to network 'A' and QoS/Cost. (The shape of curves are similar but the values are not.)](image)

The single network matrix and the derived parameter \( m; g_H; \) etc. differentiate between the two networks via the stationary distribution of the Mobile Nodes. If the \( \eta \) is high then handover rate from Network 'A' to 'B' is much higher than backwards. This means that the MNs spend more time at the Access Points of Network 'B'. When I calculate \( m \) we multiply the shortest paths in the network graph with this probability distributions thus paths with higher Utility (the paths of Network 'B') will have bigger weights that says we use them more often.

### 2.3.4 Compare the battery consumption of different approaches

Mobile devices are powered from batteries, which capacity is severely restricted due to constraints on size and weight of the device. For this reason energy efficiency and optimal management of power consumption of these devices is very important to their usability.

My mobile management framework is able to handle the battery consumption as well and compare the different approaches in this point of view. Let me introduce the \( C_{BATTERY} \), which denotes the battery cost of the mobile node. More, different paper have been proposed in the literature to cope with the battery consumption of smartphones components [75][76][77]. The table 4 shows some energy consumption distribution based on typical usage patterns. Based on these \( C_{BATTERY} = C_{com} + C_{CPU} + C_{screen} + C_{other} \).

| Table 4. Energy consumption distribution |
One can see that much depends on the workload, because of this I handled this problem general (eq. 15). From the point of position management view only $C_{\text{com}}+C_{\text{CPU}}$ is significant, the consumption of other components (for example screen) are not affected. I introduced a derived cost function $C_{\text{BATTERY}}^*$:

$$C_{\text{BATTERY}}^* = C_{\text{com}}+C_{\text{CPU}} = C_{\text{AIR}} + f_{\text{MNrel}}(C_{\text{PROC}})$$  (15),

where $C_{\text{AIR}}$ is the air interface usage cost determined earlier, and $f_{\text{MNrel}}$ is a function, which gives the $C_{\text{PROC}}$ mobile related part. Used this equation the battery cost of the different position management approached can calculated or compared to find the best energy saving solution. When a position management strategy saves more energy compared to another one, it depends on the followings:

- network configuration ($n$)
- technology and cost constants ($c$)
- mobility ratio ($\rho$)
- workload (Table 4.)

I compared two classical mobility solutions, MIPv4 and LTRACK. I used the More meshed network configuration ($n$) introduced in section 2.3.1 and the technology and cost constants ($c$) determined in section 2.3.2.

![Figure 16](image-url)  

**Figure 16:** One the left hand side summarized battery cost functions ($C_{\text{BATTERY}}^*$) of MIPv4 (solid) and LTRACK (dashed) with the vary of the mobility ratio: $\rho$. On the right hand side the percentage difference between the two approach.

In Figure 16 one can see how the battery cost changes of MIPv4 and LTRACK with the mobility ratio ($\rho$). It is clear that with the bigger frequency of handovers ($\rho$) the wireless tracking solution (LTRACK) consumes more energy because of the increasing number of
position updates. To determine an accurate difference between the two approach, let me set the mobility ratio to 0.7. In this case the battery usage of LTRACK is higher with 10.04% and taking into account the casual workload from Table 4 the battery consumption gain is 6.7

Using the above described methodology battery consumption is comparable with different configuration and approaches by MMM.

2.4 Overview

In this main section I grabbed numerous significant parameters of mobility and proposed a framework to model the mobile node behavior and the network independently of the very technology used. As an example I modeled some general management strategies and obtained technology constants with simulations to give some analytical results. I showed how each modeled protocol responds to tendencies in technology developments and trends and proposed a method to deal with vertical handover decisions using the framework. With my work it can be identified the mobility management that gives the best solution in different given network scenarios and which aspect of resources could be a bottleneck in each case. This allows operators to tune the parameters of their systems, plan their networks and researchers to propose mobility management solutions of low cost.
3 General mobility modeling and location prediction based on Markovian approach – MMCS-Markov Movement model Creator System

In wireless networks one of the most important tasks is modeling the users’ movement. A precise model based user movement prediction supports network planning, self-configuring, self-healing networks, fraud detection an admission control improving the quality of service (QoS).

The different movement models in the literature can be divided into two major groups, individual and group movement models. There are diversified mathematical solutions in both groups, from which the most popular is the Markov model.

As I introduced in previous section the aggregated movement behavior of the Mobile Nodes can be modeled with a finite state continuous Markov chain and the chain is given with a rate matrix $B_Q$. But instead of a simple aggregated handover rate a complex and efficient movement model is appropriate for a more sophisticated network model.

In this section I introduce a precise Markov mobility model based on network parameters, to capture the additional information contained in the user location traces. My goal is to provide a synthetic model capable of capturing the unique properties of specific locations, e.g. urban areas, such as crowded parks, one-way streets etc.

3.1 Mobility models in the literature

Different mobility models have been proposed in the literature to cope with user mobility in different wireless and mobile networks (e.g. cellular networks, ad-hoc networks etc.) In this chapter I give a short overview on the most widely employed mobility models.

3.1.1 Mobility models

In the Random Walk mobility model, [13] the node moves from its current location to a new location by randomly choosing a direction and a speed. The Random Walk model defines user movement from one position to the next with memoryless randomly selected speed and direction. Each movement in the Random Walk Mobility Model occurs in either a constant time interval $t$ or a constant distance traveled $d$. When a mobile node reaches a simulation boundary, it simply bounces off the simulation border with an angle determined by the incoming direction, then the node continues along this new path. Many derivatives of the Random Walk Mobility Model have been developed including the one, two, three - and d-dimensional walks.

Chiang’s mobility model [63] defines the Probabilistic Version of Random Walk model, which utilizes a probability matrix to determine the position of a particular mobile network in the next time step. The position of the node is represented by three different states for position $x$ and three different states for position $y$. In this mobility model state 0 represents the current ($x$ or $y$) position of a given node, state 1 represents the node’s previous ($x$ or $y$) position, and state 2 represents the node’s next position if the node continues to move in the same direction. The probability matrix used in the probabilistic random walk model, where each entry $P(a,b)$ represents the probability that a node will go from state $a$ to state $b$. The values within this matrix are used for updates to both the node’s $x$ and $y$ position.
In the Random Waypoint Model [64] the user stays at a particular location for a specified time period before moving on to the next in a randomly chosen direction with speed uniformly distributed between zero and maximum speed. The model derived from the Random Walk model breaks down the entire movement of the user into a series of pause and motion periods.

A density wave is the clustering of nodes in one part of the simulation area. The Random Direction Mobility Model [65] was created to overcome density waves in the average number of neighbors produced by the Random Waypoint Mobility Model. In the case of the Random Waypoint Mobility Model, this clustering occurs near the center of the simulation area.

The Modified Random Direction Mobility Model is a slight modification to the Random Direction Mobility Model [65]. In this modified version, the node continues to choose random directions but they are no longer forced to travel to the simulation boundary before stopping to change direction. Instead, a node chooses a random direction and selects a destination anywhere along that direction of travel. The node then pauses at this destination before choosing a new random direction.

In the Boundless Simulation Area Mobility Model, a velocity vector \( v = (v, \Theta) \) is used to describe an node’s velocity \( v \) as well as its direction \( \Theta \). In this model a relationship between the previous direction of travel and velocity of a node with its current direction of travel and velocity exists [73]. Both the velocity vector and the position are updated at every \( \Delta t \) time steps according to the following formulas:

\[
\begin{align*}
  v(t+\Delta t) &= \min[\max(v(t)+\Delta v), V_{\text{max}}], \\
  \Theta(t+\Delta t) &= \Theta(t) + \Delta \Theta, \\
  x(t+\Delta t) &= x(t) + v(t) + \cos \Theta(t), \\
  y(t+\Delta t) &= y(t) + v(t) + \sin \Theta(t).
\end{align*}
\]

Here, \( V_{\text{max}} \) is the maximum velocity, \( \Delta v \) is the change in velocity which is uniformly distributed between \([ -A_{\text{max}} \Delta t, A_{\text{max}} \Delta t ] \), and \( A_{\text{max}} \) is the maximum acceleration of a given node. The parameter \( \Delta \Theta \) means the change in direction which is uniformly distributed between \([ -\alpha \Delta t, \alpha \Delta t ] \), and \( \alpha \) is the maximum angular change in the direction.

To adapt different levels of randomness, one can use the Gauss-Markov Mobility Model. In this model initially each node is assigned a current speed and direction. In this mobility model at fixed intervals of time, \( n \) movement occurs by updating the speed and direction of each node. The value of speed and direction at the \( n \)-th instance is calculated based upon the value of speed and direction at the \((n+1)\)-st instance and a random variable using the following equations:

\[
\begin{align*}
  s_n &= as_{n-1} + (1-\alpha)s + \sqrt{(1-\alpha^2)s_{n-1}}, \\
  d_n &= ad_{n-1} + (1-\alpha)d + \sqrt{(1-\alpha^2)d_{n-1}},
\end{align*}
\]

where \( s \) and \( d \) are the new speed and direction of the node at time interval \( n \), and \( \alpha = 0 \) is the tuning parameter used to vary the randomness, while \( s \) and \( d \) are constants representing the mean value of speed and direction. The parameters \( s_{n-1} \) and \( d_{n-1} \) are random variables from a Gaussian distribution. Totally random values are obtained by setting \( \alpha = 0 \) and linear motion is obtained by setting \( \alpha = 1 \) [66], while intermediate levels of randomness are obtained by varying the value of \( \alpha \) between 0 and 1.
In the City Section Mobility Model it is represented a section of a city where the ad hoc network exists [67]. The streets and speed limits on the streets are based on the type of city being simulated. The streets might form a grid in the downtown area of the city with a high-speed highway near the border of the simulation area to represent a loop around the city.

If a flexible mobility framework for hybrid motion patterns is needed, one can rely on the Mobility Vector Model [68]. A mobility vector expresses the mobility of a node as the sum of two sub vectors: the Base Vector \( B=(bx_v, by_v) \) and the Deviation vector \( V=(vx_v, vy_v) \). The base vector defines the major direction and speed of the node while the deviation vector stores the mobility deviation from the base vector. The mobility vector \( M \) is expressed as \( M=B+\alpha V \) where \( \alpha \) is an acceleration factor.

The location history traversed by a mobile user is exploited in High-Order Markov Model that is described in [17][21]. The model focuses on the identification of a group of especially harmful internal attackers. The order-o Markov predictor assumes that the location can be predicted from the current context, which is the sequence of the previous o most recent characters in the location history.

F. Lassabe et al. present a mobility model adapted to the logging of mobile positioning or to the tracking of mobiles. This model is based on the All-Kth Markov Model [23]. They present two predictive models from the AKMM: the K-to-I past Model and its improvement, the K-to-I past* Model. The model defines a Markov state-space constructed of the possible user trajectories. Each state describes a trajectory section of 1 to K previous locations. The model predicts future locations based on the possibilities of each transition between states. A threshold value is used to select a group of locations which are likely to be visited in the next step, so a handoff procedure can be prepared for each one.

Shiang-Chun Liou et al. present a mobility model with two-tier cell structure in [28]. The user trajectory is defined based on the logical function of velocity, direction, acceleration and position. This logical function is converted to a model that uses three preceding geographical locations to estimate the fourth parameter. The location prediction with this estimation enables the network operator to make preparations for a future handoff in the group of cells that are likely to be crossed. Two-tier cell structure is used to decrease the waste of bandwidth due to reserved resources of a future handoff. The two tiers can be described in a mobile cell as a function of distance from the base station (first tier). While the mobile node is close to the base station, it is unlikely that the even with a sudden trajectory modification the mobile node steps into another cell. On the other hand, if the mobile node is more close to the cell boundary (second tier), the possibility of a handoff is increasing.

W. Ma et al. proposes a user mobility pattern (UMP) based model (Mobility Pattern-Based Scheme – MPBS) in [29]. The MPBS is a general method to follow users in the network without expensive paging operations if the user meets some requirements. The model defines a personal mobility pattern list which consists of a sequence of register areas (RA, i.e. mobile cells), and a time-sequence of the trajectory on the RA sequence. The time-sequence is built up by the timestamps of handoffs between RAs, and the dwell times for each RA. Based on the time- and RA-sequence an exact timeline can be defined which is followed by the user. The operator does not need to page the user in different RAs, because the timeline shows that in which RA is the user located at the actual timestamp. Naturally, the ideal user who always follows the timeline does not exist, but the time-sequence and RA-sequence provides information even if the actual timeline differs from the prerecorded one. A categorization is
presented with four categories where the first category is the ideal user with a timeline-compatible trajectory. Second category involves users who are following the RA-sequence but with time delays or hurries, that is the network operator can find the user in the remaining RA set after the last paging or location update. Users who are located in the appropriate RA set, but are not following the sequence are in the third category. The fourth category is for the users who are located out of their UMPs, that is their actual trajectories are not close the prerecorded ones.

3.1.2 Group mobility models

In mobile networks there are many situations where it is necessary to model the behavior of nodes as they move together. The most general of these group models is the Reference Point Group Mobility (RPGM) model. Specifically, three of the group mobility models (Column, Nomadic, and Pursue) can be implemented as special cases of the RPGM model.

In the Reference Point Group Mobility (RPGM) model, [74] the motion of the group center completely characterizes the movement of its corresponding group of nodes. The RPGM model represents the random motion of a group of nodes as well as the random motion of each individual node within the group. The logical center for the group is used to calculate group motion via a group motion vector $GM$. In this mobility model the individual nodes randomly move about their own pre-defined reference points, whose movements depend on the group movement. When the updated reference points $RP(t+1)$ are calculated, they are combined with a random motion vector $RM$, to represent the random motion of each node about its individual reference point.

This Column Mobility Model [69] represents a set of nodes that move around a given line or column, which is moving in a forward direction. The model useful for scanning or searching purposes and a slight modification of the Column Mobility Model allows the individual nodes to follow one another. Each node is placed in relation to its reference point in the reference grid; the node is then allowed to move randomly around its reference point via an entity mobility model.

In the Nomadic Community Mobility Model, each node uses an entity mobility model like the Random Walk Mobility Model, to roam around a given reference point. The model represents groups of nodes that collectively move from one point to another [70]. However, within each community or groups, the individuals move in random ways. When the reference point changes, all nodes in the group travel to the new area defined by the reference point and then begin roaming around the new reference point.

The Pursue Mobility Model [70] represents nodes tracking a particular target. The model consists of a single update equation for the new position of each node. The random vector value is obtained via an entity mobility model like the Random Walk Mobility Model. The amount of randomness is limited in order to maintain effective tracking of the node being pursued. The model combines the current position of a node, a random vector, and an acceleration function to calculate the next position of the node.

3.1.3 User-distribution prediction models

The Shadow Cluster Scheme [71] estimates future resource requirements in a collection of cells in which a mobile is likely to visit in the future. The shadow cluster model makes its prediction based on the mobile’s previous routes. In this model, the highway traffic with
various constant speeds are simulated and users travel in forward and backward directions. The shadow cluster model improves estimation of resources and decision of call admission. In the study by Chao and Chen, user mobility is estimated based on the aggregate history of handoff observed in each cell. Shadow Cluster Concept takes its prediction, based on the mobile’s previous routes.

The Sectorized Ad-Hoc Mobility Prediction Scheme [72] achieves maximum accuracy in movement prediction. In this model the prediction process should be restricted to areas of high cluster change probability. The sectorized ad hoc mobility prediction scheme makes use of the cluster-sector numbering scheme to predict user movements in an ad hoc network.

In the cluster based [72] model the cluster head has complete knowledge of each of its member nodes. In this model the location of the user is defined with respect to its position with that of the cluster head. Assuming a circular cluster structure there is a region of the cluster in which all the nodes belonging to the cluster are in closest proximity to each other. All nodes in this region of the cluster are within communication range of each other, and the nodes in this region of the cluster will not satisfy the requirements for membership to any of the neighbouring clusters.

Sándor Szabó proposes a Ring-Based Mobility Prediction and resource reservation algorithm in [27]. A cell cluster is divided into three cell groups, where the first group is equivalent to the central cell of the cluster, the second and third groups are consist of the cells that are located in the first and second cell ring around the central cell respectively. The pre-handoff resource reservation is derived from the possibilities of the event that the mobile node steps from the central cell into a cell of the second or third cell-group. This approach can be considered as the generalization of the two-tier cell structure described in [28] to an inter-cell level.

3.1.4 Markov model

I have chosen the Markov model as mathematical model to handle user movement in cellular mobile network, because it is widely-used model and a good, easy way to model sets of user trace information.

Markov model is a stochastic, mathematical model in probability theory. Markov-chain is the most simple representation of the Markov model. It models the state of a system with a random variable that changes through time. In this case the Markov property suggests that the distribution for this variable depends only on the distribution of the previous state. Markov model is a widely-used mathematical solution for telecommunication problems [60]. I used discrete-time, finite-state and infinite-state Markov-chain to model the mobile movement.

3.2 Classifying Markov mobility models

I investigated the Markov models that can be found in the literature and based on these I extended the model introduced in the previous chapter with specific parameters describing the user movement.

Thesis 2.1 I have created a classification method for the discrete-time Markov movement models (MMCS – Markov Movement-model Creator System). Every M model which states express the staying in an area is definable exactly in \( M< L, R, O > \) form, where \( L \) is 'level', \( R \) is 'resolution' and \( O \) is the 'depth' of the model. The solution is able to compare and to analyze the efficiency of Markov movement models in different network scenarios [J1,J2].
3.2.1 Deriving user movement related network parameter of given network for MMM

User movements in a cellular network can be described as a time-series of radio cells the user visited. The handover event of active connections (e.g. cell boundary crossing) is recorded in the network management system’s logs, thus the information can be extracted from the management system of cellular mobile networks, such as GSM/UMTS/LTE networks. The users movements is described by the dwell-time and outgoing probabilities (the probability of a user leaving for each neighboring cell). These parameters can be calculated for each cell based on the time-series of visited cells of the users. However, in some cases, these two parameters – dwell-time and outgoing probabilities – are not enough to capture all the information in the time-series of user movements. In many situations, the outgoing probabilities are correlated with the incoming direction of the users, so the movement contains memory.

After the basic notations introduced in previous main section a method is introduced to process the network traces. I introduce a method to process the network traces to calculate typical parameters of the mobility. The trace entry describes the events in a cellular network. An event might be a status change of the given user (e.g. mobile node is idle state, voice call or data transfer is set up, cell boundary crossing). The logical location of the event is determined with the MAP ID where the user is located at the timestamp of the event. (Table 5). The events are recorded in the network management system’s logs, thus the information can be extracted from the management system of cellular mobile networks.

<table>
<thead>
<tr>
<th>TIMESTAMP</th>
<th>USER ID</th>
<th>CELL ID</th>
<th>STATUS (OR EVENT)</th>
</tr>
</thead>
<tbody>
<tr>
<td>09:21:43:12</td>
<td>41</td>
<td>4951</td>
<td>Change to Idle</td>
</tr>
<tr>
<td>09:21:45:48</td>
<td>42</td>
<td>4957</td>
<td>Change to Idle</td>
</tr>
<tr>
<td>09:21:49:21</td>
<td>41</td>
<td>4957</td>
<td>Voice Call</td>
</tr>
<tr>
<td>09:21:50:38</td>
<td>19</td>
<td>5341</td>
<td>Sending/Receiving Data:Traffic class 2</td>
</tr>
<tr>
<td>09:21:51:42</td>
<td>84</td>
<td>7120</td>
<td>Change to Idle</td>
</tr>
<tr>
<td>09:21:51:58</td>
<td>19</td>
<td>5348</td>
<td>Sending/Receiving Data:Traffic class 2</td>
</tr>
<tr>
<td>09:21:52:26</td>
<td>19</td>
<td>5348</td>
<td>Change to Idle</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The aim is to derive the parameters of the user mobility, therefore I should pick the relevant entries from the network trace. In my work I focused on location changes of users, handovers, and initializing or receiving calls. These events are observed during a time interval that is considered to be the reference interval for deriving model parameters.

I assumed that the user distribution in the network is given at the first moment of the reference interval. I created a discrete sample series where samples are taken with \( \Delta t \) time, that is a location status is assigned to every users per \( \Delta t \) time. \( \Delta t \) is defined system-wide as the minimum of the time interval elapsed between two events registered to the same user. That is the sample frequency is set to the “fastest” user in the network. This ensures that every user event and all status reports are processed. With this sampling a MAP ID and a status can be determined to every user in every timeslot. The sampling results an \( n_u \times n_T \) sized \( P \) matrix, where \( n_T \) denotes the number of timeslots, and \( n_u \) the number of users in the model. \( P \) matrix stores the MAP IDs and status of each user in each timeslot.
The relative frequency of any status can be determined based on $P$ matrix, for instance the relative frequency of receiving voice call in a MAP, or even the handover rate between two different MAPs. I defined the $S$ set, which contains all possible statuses and events appearing in the logs. The important ones are the following:

- receiving voice call
- receiving data call
- initializing voice call
- initializing data call
- fall back into idle status

Depending on the detail of the logs and on the requirements more or different statuses, events could be investigated as well. For example if more data traffic classes are identified in the logs, then they could also be differentiated. But for us the above mentioned statuses are sufficient in the model.

To determine the relative frequencies of statuses in a MAP, the status must occur frequently enough, otherwise it is neglected. Let me define the $n \times n$ matrix, $C = [c_{i,j}]$, where $c_{i,j}, s \in S$, is the occurrence of status with user $i$, in MAP $j$ from the $P$ matrix. The average occurrence of $s$ status in the network is:

$$c^s = \frac{\sum_{i=1}^{n_u} \sum_{j=1}^{n_m} c_{i,j}}{n_u \cdot n_m}. \quad (16)$$

The parameter $c^s$ can be used as a main requirement in order to create valid model based on the network. So the network must be monitored for sufficient time before I create a model from it.

Parameter $\epsilon_c$ denotes the minimal occurrence of a status for acceptance, if the occurrence is smaller than $\epsilon_c$, the rate will be 0. Based on this the relative frequency matrix of a status can be determined as the following:

$$D^s = [d^s_{i,j}] = \begin{cases} 0, & c^s_{i,j} < \epsilon_c \\ c^s_{i,j} / c^s_j, & c^s_j = \sum_{vi} c^s_{i,j} , s \in S. \end{cases} \quad (17)$$

In fact $d^s_{i,j}$ is the probability of getting into the status $s$ happens to user $i$ in MAP $j$.

The rate of receiving call is introduced: $\mu$. It can be determined for every MN in every MAP from the $D^s=[\mu_{s,i}]$, but the average value is also calculable in similar manner described above:

$$\mu = \frac{\sum_{i=1}^{n_u} \sum_{j=1}^{n_m} \mu_{i,j}}{n_u \cdot n_m}. \quad (18)$$

Let us have the corresponding network graph given with its weighted adjacency matrix: $A$. Let us assume that the aggregated behavior of the Mobile Nodes can be modeled with a finite state continuous Markov chain (the handover or call arrival rate is a Poisson process with various intensity parameters as in many works, e.g. [2]). The chain is given by a rate matrix $B_\theta = [b_{ij}]$. In this matrix, all the possible MAP-s are listed so the matrix will be a $n_m \times n_m$ matrix.
where each element $b_{ij}$ denotes how frequent the movement of the mobile is from MAP$_i \rightarrow$ MAP$_j$. If an MA is not a MAP then there are 0 values in its row and column (i.e. we treat it the same way that the MN cannot or never attaches to it). From the rate matrix the transition matrix $B_{ni}$ can be determined easily. We assume that the matrix $B_{ni}$, without the non-MAP nodes, is practically irreducible and aperiodic that implies that the chain is stable and there exists a stationary distribution. This will be denoted by a density vector $b$. Other $B$ matrices can be determined for a single user, user group or all the users as well and they can be assigned to a state in the network also. According these assumptions, for example $B^{i,ns}$ is the transition probability matrix of user $i$, when it is in state $s$. Another network describing parameter, which is useful during the modeling, is the number of visited MAPs by the user or users. That is calculated as follows:

$$n_{vm} = \frac{1}{n_u} \cdot \text{sign}(B_{ni} \cdot \mathbf{1}).$$  \hfill (19)

A general network describing parameter, the weighted average of visited MAPs is $w_{vm}$ (eq. 20). The other parameter is the average number of neighboring MAs (eq. 21) that can be accessed via a wire from a given node: $w_{vm}$. It should be also weighted with the probability density of the MN. These parameters are the followings:

$$w_{vm} = \frac{\sum b \cdot \text{sign}(B_{ni}) \cdot \mathbf{1}}{n_u}, \hfill (20)$$

$$w_{nm} = \frac{(b \cdot \text{sign}(B_{ni})) \cdot \mathbf{1}}{n_u}. \hfill (21)$$

When talking about an existing network, the parameters described in this chapter can be calculated easily, producing the base of the model.

### 3.2.2 Classification

In this chapter, the simple classification of motion models are presented. The aim is to compare different models easily and to analyze in variant network environment. The analysis is to determine the attributes of the individual models, the level, the depth and the resolution and hence the models can be rated in a general marking system.

The classification system handles simple, general Markov-chain based mobility models, in that a user or users can be located in different Markov states. MAP or group of MAPs (or merged MAPs based on a special relationship) is mapped to a state or more states of the Markov-chain model. Let us define $X(t)$ random variable, which represents the movement state of a mobile terminal during timeslot $t$. The transition probabilities of the Markov model can be determined from the describing parameters of the network (see Chapter 3). Let us assume that the Markov chain is always irreducible and aperiodic, so the stationary user distribution is determinable.

Two main types of these Markov mobility model is distinguished, User-Centralized and Access Point-Centralized. The latter one is further separated into two subtypes. Figure 17 depicts this main classification.
Besides the simple classification I have defined attributes describing the main characteristics of the models. In the next subsection these groups of models and the attributes are explained.

### 3.2.3 User-centralized Markov models (UCM)

A user or a group of users from the network is selected for observation in user-centralized Markov models. The users’ movement behavior is modeled with a Markov model. Only the MAPs which are visited by the selected user(s) are taken into account, other MAPs, and other users do not affect the structure of the movement model.

Figure 18 shows how to represent a user centralized model. The chosen user in the example visits only the MAPs between ID 1 and ID 5. Each MAP is mapped into one standalone Markov state. This is a very simple model, where the stationary distribution of the Markov chain is equal to density vector \( b \) as the stationary distribution of \( B_{ij} \) transition matrix.

This usage of this model is reasonable, if the behavior of the user is to be investigated, or a user profile is needed for example for fraud detection.

Most of the Markov mobility models in the literature belong to this class, for example [17],[22],[23].
3.2.4 Access Point-centralized Markov models (ACM)

The access point-centralized Markov models can be used when the user distribution in a selected MAP or group of MAPs must be determined. Instead of modeling the behavior of an individual user, a MAP and its environment is to be observed. In these cases a MAP or more MAPs and defined neighbors are selected according to a requirement. The users who stepped into the area of the observed MAPs are investigated and their distribution is used to build a model for prediction.

Two guidelines exist:
- In the structured model for a predefined reason certain MAPs are grouped together, which creates a regular structure in the model.
- A MAP or MAPs are simply mapped into a state of Markov model. This method is called unstructured model.

Details and examples are presented in the next sub-chapters.

3.2.4.1 Structured Markov Models (ACSM)

In the structured Markov model groups of MAPs are defined. The grouping can be derived from user behavior, geographical specialty or even from network requirement. Figure 19 shows examples for structured solutions, the Ring Model (RM) [27] and the M3 model [26].

In the RM (Figure 19.b) the ring consists of cells surrounding a central cell. The concept is to simplify the calculations, if we are interested only in the number of users arriving to a given ring, or leaving a given ring during a time period. Internal movements are disregarded [27].

The M3 model handles users in four different Markov state, the right-area, left-area, stay and outside state. More details about the M3 model are in section 3.5.

![Figure 19.a: The creation of access point-centralized/structured models, M3 model [J2]](image)
Let me introduce the theoretical error $E_T$, which is the summarized and standardized wrong predictions by MAPs. In detail during the prediction, if we want to determine the number of users in the MAPs of the group (for example the right-area state in M3, or the first ring in RM), then the predicted number of the users for the group is distributed uniformly between the MAPs in the group. Obviously this step brings a theoretical error ($E_T$) into the prediction process. For example in the left area state of the $M3$ model there are MAP$_1$, MAP$_2$ handled together as a group. We know that in the left area state 100 users move. For lack of further information 50-50 users are predicted in MAP$_1$, MAP$_2$. Actually there is 25 users in MAP$_1$ and 75 in MAP$_2$. In this case the $E_T$ is 50% in MAP$_1$, MAP$_2$ as well.

### 3.2.4.2 Unstructured Markov Models (ACUM)

If we try to predict the user’s distribution in a city having irregular, dense road system, or in a big park where people are able to move around then the handover intensities could differ thus the calculations above could produce errors. From this point of view the best way is if we represent all of the neighbour MAPs as a separated Markov state, so this results the Unstructured Markov Model. The results from determined stationary distribution are easy to map back, into the MAPs.
Figure 20: Access point-centralized/Unstructured/M7 model [J1]

Figure 20 depicts the methodology of unstructured model representation. The Markov chain in Figure 20 is similar to M3 model. In this M7 model all of neighbour MAPs are mapped into Markov-chain states. The M7 and generalized Mn model described later, in 'Markov model examples' section.

3.2.5 Attributes

The example models introduced above are the simplest ones in their class. I mentioned earlier the attributes which describe the main characteristics of the Markov models. Supported by these attributes more complex, more sophisticated models could be classified or constructed for solving more difficult problems.

The examples mentioned in the introduction of this section used present only one attribute at a time to keep the simplicity and distinctness. Of course the attributes could be used together in any number and combination.

3.2.5.1 Level of the model

As mentioned in Chapter 3.2.1, \( B_{II} \) could be determined from the \( P \) matrix for every status as well. Models can treat users or user groups separately introducing a new dimension, new level into the given movement model. This is justified when:

- certain users’ movement significantly differs from the usual
- we would like to distinguish the users with for example a multi-level movement model (for example different admission control strategy (CAC) is used for the users in voice call, then the users downloading data from the internet)

In these cases the \( B_{II} \) must be calculated for different statuses. This diversity in the model is represented by 'levels' or 'dimension' (Figure 21). The transition rates between the levels show the intensity status changes in the current MAPs. This model is similar to the one described in [61].
Figure 21: Attributes in Markov mobility model classifying: Example for meaning of ‘level’

The number of levels in the model is determined with $n_L$. A level is denoted with $L$, the levels in the model are marked with $L$ vector, where $L = [L_1,...,L_n]$. A specific $L$ is based on its $B_{II}$ matrix.

3.2.5.2 Resolution of the model

The model’s resolution describes if we join two or more adjacent MAPs and represent them with one state in the Markov chain. This usually happens when the given MAPs shouldn’t be handled separately due to the fact that users behave similarly within these or simply a minimal number of users are connected. The similar user behavior could be identified of outgoing predictions of users in two adjacent MAPs. The joined MAPs handled henceforward as a new major MAP.

By this the complexity of the model can be decreased. Figure 22 shows an example, in which the 6 neighbor MAPs of an access point-centralized, unstructured model are merged into 3 new major MAPs.
Figure 22: Attributes in Markov mobility model classifying: Example for meaning of ‘resolution’

‘Grouping’ explained in Chapter 3.3.2.1 (structured models) is not equal to ‘merging’ mentioned here. As the result of ‘merging’ new, major MAPs are created instead of the initial ones. A ‘grouping’ organizes the MAPs into a structure.

Every level could have its own resolution. The resolution is denoted with $R$, $R = \{G_1, \ldots, G_m\}$, where $G$ is a set of merged MAPs, and $n_M$ the number of new MAPs after merging. The $R$ is described with a general rate, $n_m: n_M$. The vector $R = [R_1, \ldots, R_m]$ contains the resolution rules to every level.

### 3.2.5.3 Memory of the model

The application of the recent user locations has a crucial importance in a variable, directional user motion. Neglecting the preceding transition series of a user in the MAP results that the estimation could work with a theoretical error ($E_T$, like in M3 and RM model) [26].

It is very important that the usage of this type of ‘memory’ does not violate the Markovian property, the memorylessness is still true for the Markov model created by the MMCF.

I show a simple example which represents the effect of depth or memory in the model. If I consider two roads shown is Figure 23.b, the accuracy of the transition probability estimations is higher when the model knows where the users come from than an estimation which cannot distinguish the users on the two roads (Figure 23.a.).
To clarify the error rate of a memoryless model compared to an algorithm that calculates with memory. Let me define the following parameters:

- the number of incoming users on the upper road at timeslot \( t \) is \( \text{in1}_t \),
- the number of incoming users on the lower road at timeslot \( t \) is \( \text{in2}_t \),
- similarly the number of users leaving on the upper road at timeslot \( t \) is \( \text{out1}_t \),
- the number of users leaving on the lower road at timeslot \( t \) is \( \text{out2}_t \),
- and the user movement directions with a simple transition matrix \( P = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix} \).

The model in Fig. 23.a calculates the number of leaving users (\( \text{out1}, \text{out2} \)) with a historical estimation of the \( P \) matrix, and with the sum of \( \text{in1} \) and \( \text{in2} \) (the total number of users in the observed cell), but without the knowledge of \( \text{in1}, \text{in2} \).

In other (Figure 23.b) algorithm, the use of \( \text{in1}, \text{in2} \) means that the model calculates based on the information of the income users. The income users can be considered uniform or different (marked users). Since the latter is more accurate, in this comparison I use marked users.

Let me assume that the historical estimation of the first model’s \( P \) matrix is based on the previous timeslot. That is the \( P(\text{out1}_{t-1}) \) and \( P(\text{out2}_{t-1}) \) probabilities are \( \text{out1}/(\text{out1}+\text{out2}) \) and \( \text{out2}/(\text{out1}+\text{out2}) \), respectively. Applying the same assumption on the algorithm with memory, the number of leaving users can be calculated with the \( P \) matrix itself, that is \( \text{out1}_{t+1} = \text{in1}_t * P_{11} + \text{in2}_t * P_{21} \) and \( \text{out2}_{t+1} = \text{in1}_t * P_{12} + \text{in2}_t * P_{22} \). At a given and constant \( P \) matrix let me assume that the income user distribution varies, that is the \( \text{in1}/\text{in2} \) ratio (Income Distribution - ID) changes. Fig. 24. shows the error of HOV compared the estimation using memory.

![Figure 24: Prediction error in percents –](image)

Given \( \text{in1}/\text{in2} = 1 \) and \( P = \{0.75, 0.25\} \{p_{21}, 1-p_{21}\} \), \( p_{21} \) plotted with four different values

The 'memory-less' model works with error if \( ID_{t-1} \) is different from \( ID_t \) which is caused by the fact that the historical \( P \)-estimation in this special case equals number of leaving users of the previous timeslot. That is it does not include the actual \( ID_t \) value. Contrarily the memory-model calculates with the actual number of income users and the \( P \) matrix itself, which results the exact probabilities of the leaving users distribution. The error rate caused by the lack of memory increases as the variance of ID increased that is \( \text{in1}/\text{in2} \) ratio changes.

Using memory cannot enhance furthermore the accuracy of the estimation if \( p_{21} = p_{11} \) and Fig. 24. shows a constant zero error rate (\( p_{11} = p_{21} = 0.75 \)). In this case the leaving direction of
each user is independent of the income direction and the memory is useless, since users arriving from each direction are leaving towards a given direction with the same probabilities. The results show that my proposition of using memory in a mobility model significantly increases the accuracy of the model in cases when the ID distribution in an arbitrary cell has high variance, or has periodicity without stationary distribution.

Therefore, an o-depth could be determined for my Markov models similar like in [17]. In my model, sequence of MAP IDs can be assigned to every MAP not to a user; \(ID_1, ID_2, \ldots, ID_n, \ldots\), where \(ID_i\) denotes the identity of the MAP visited by the mobile before it stepped into the current MAP. The last element of the sequence is always the current MAP. The future locations of the mobile in most of the cases are correlated with its movement history. The probability that the user moves to a particular MAP depends on the location of the current cell and a list of cells recently visited. If only the current cell is taken in account, like in previous examples, the depth is 1.

For every MAP different depth could be assigned, which determines the length of the recently visited MAP ID list before the current MAP. Since a MAP could be reached on different paths by the users, therefore more specific MAP ID list could belong to a MAP, and for this reason more Markov state assigned to a MAP, see Figure 25.

![Figure 25: Example for meaning 'memory', 2nd model [23]](image)

Thus \(O = [a_1, \ldots, a_n]\) matrix denotes the depth of the model for a level, where \(a_i\) is the applied sequence length of previous visited MAPs to MAP. Generally the \(n_m \times n_L\) \(O\) matrix (\(O = [O_1, \ldots, O_m]\)) belongs to a Markov model.

The weighted average depth for a model is the following:

\[
W_O = \frac{\sum_{i=1}^{n_m} b^i \cdot O^i}{n_L}. \tag{22}
\]
3.2.6 Complexity

The complexity of the model could be denoted with the number of states. Following the determining of attributes the number of states is:

\[ n_{\text{states}} = \sum_{\forall i \in \mathbb{L}} n_i' / n_M' \cdot w_{\text{room}} w_o. \]  \hspace{0.5cm} (23)

3.2.7 Performance analysis

In this chapter I have carried out an analytic cost evaluation using some Markov models found in the literature with different network configurations. The models are examined in different network environments. In every scenario the analysis is applied to a cluster of 7 hexagonal radio cells. (Figure 26). In my interpretation the performance of a mentioned Markov models depend on the theoretical error (\( E_T \)) and the number of states (\( n_{\text{states}} \)), in this case the special theoretical cost were determined as function of these two parameters: \( C_{\text{MM}}=f(n_{\text{states}}, E_T) \). The lower this cost is, the better the performance. The computational capacity increases, hence the steady state probabilities determination of Markov models with more state are even less difficult. But the theoretical error in prediction is more important. For these reasons I calculate the theoretical cost this way:

\[ C_{\text{MM}} = \log n_{\text{states}} + E_T \]  \hspace{0.5cm} (24),

where \( n_{\text{states}} \) is the number of states and \( E_T \) is the theoretical error. In previous sections, in case of RM, M3 and o-th models the \( E_T \) was introduced, is the summarized and standardized wrong predictions by MAPs. This is implied by handling certain MAPs joined in the model or not taking into account previous steps with sufficient depth (RM, M3, o-th) and during the prediction we calculate the users in the MAPs with linear distribution.

The following models were examined in the performance analysis:

- RM: The model introduced in chapter 3.2.5.1 (Figure 19.b) with the difference that only one ring is around the central MAP, so only states \( S, R1, \) and \( O \) exist. The predicted number of the users in \( R1 \) state is distributed uniformly between the \( MAP_r-MAP_e \).

- M3: Also introduced in chapter 3.2.5.1 (Figure 19.a.) More details can be found in chapter 3.5. \( MAP_2, MAP_3 \) and \( MAP_7 \) belong to the left-area state (L). The right-area state (R) includes \( MAP_e, MAP_2, \) and \( MAP_6 \).

- M7: The model presented chapter 3.2.5.2 (Figure 20).

- 2\textsuperscript{nd}: The extension of model M7, the memory is increased to 2. It is similar to 2\textsuperscript{nd} m Figure 25.

- 3D: This is a two-level model resulted by duplicating the Markov chains of the M7 model. The first level represents the users who establish data connections, while the second level represents the mobile users initiating and receiving voice calls.

I investigated special network environments (scenarios) to highlight the advantages and disadvantages of each model. The different network scenarios are shown in Figure 26. These are the following:

- Scenario a. : 'A park, uniform user distribution'
- Scenario b. : 'Simple road'
- Scenario c. : 'Highway to city'
- Scenario d. : 'Directional motion'
- Scenario e. : 'Differentiated users'
Figure 26: The network scenarios for performance analysis with MMCS

Table 6: The result of performance analysis with MMCS

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>RM ($n_{states}=3$)</td>
<td>0,47</td>
<td>1,81</td>
<td>1,28</td>
<td><strong>2,11</strong></td>
<td><strong>2,11</strong></td>
</tr>
<tr>
<td>M3 ($n_{states}=4$)</td>
<td>0,6</td>
<td><strong>1,93</strong></td>
<td>0,6</td>
<td>1,93</td>
<td>1,93</td>
</tr>
<tr>
<td>M7 ($n_{states}=8$)</td>
<td>0,9</td>
<td>0,9</td>
<td>0,9</td>
<td>1,9</td>
<td>1,9</td>
</tr>
<tr>
<td>2nd ($n_{states}=49$)</td>
<td><strong>1,69</strong></td>
<td><strong>1,69</strong></td>
<td><strong>1,69</strong></td>
<td><strong>1,69</strong></td>
<td><strong>2,94</strong></td>
</tr>
<tr>
<td>3D ($n_{states}=16$)</td>
<td>1,2</td>
<td>1,2</td>
<td>1,2</td>
<td>2,1</td>
<td><strong>1,2</strong></td>
</tr>
</tbody>
</table>

Table 6 shows the result of the analysis. $C_{MM}$ was calculated in every scenarios for every model. The highlighted values are the best, the boldfaced the worst results for the current scenario.

Scenario a. represents a simple park, where the distribution of the users is uniform so thus the $ET$ is 0 in each model. In this case the best performance belongs to RM, because it has the minimum number of states.

In scenario b. a road crosses the examined area. The users are distributed uniformly on the road. In the left state of the M3 model, the predicted distribution of the users in 33,3% in each MAP group (MAP7, MAP2, MAP3), while all of them stay in M7. Therefor the model has 1.3 theoretical error. The theoretical error of the RM model can be determined the same way. M7, 2nd and 3D make no error. The optimal choice is M7 in this situation.

The scenario c. presents a morning, rush-hour traffic situation towards the city. On the road the users are distributed uniformly. The road to the city located fully on the left-area state of the M3, and the road from the city is in the right-area state. Because these states are handled separately there is no theoretical error. However the RM based prediction distribute the users uniformly between the 6 neighbour MAPs.
Scenario d. is very similar to the example presented in chapter 3.2.3.6. The users from MAP$_2$ move to MAP$_4$ via MAP$_1$. The other path is MAP$_7$-MAP$_1$-MAP$_5$. 50% of users step out from MAP$_1$ to MAP$_2$, the other 50% to MAP$_5$. In the measured time-slot all of the users are in MAP$_1$, on the road below, so the next step is MAP$_5$. Because only the 2nd model takes into account the previous step, all other model commit the theoretical error.

Scenario e. is a special case. The mobile users in the office generate data traffic, in the park there are rather voice calls. At current $t$ timeslot in MAP$_1$ the park is empty, users moves only in the building. In this case 3D model makes no error as opposed to other models.

One can see which models perform well is which special situations. This implies that the certain model has to be carefully chosen or constructed for the given network location.
3.3 Markov Model Constructor Framework

In the previous chapter I showed that every model works well in a given network environment. Therefore the optimal model have to be chosen, or constructed for a specific network environment.

If we would like to create an own Markov movement model, take into account all historical data (depth) and treat all different user behaviors distinguished it will result in a model with high number of states. There for it is practical to allow a minimal error and find the Markov model with the least number of states.

In this chapter I give guidelines to construct a Markov movement mobility model for a network, and I present the determination of the classification system attributes.

**Thesis 2.2** I have developed a general, discrete-time Markov Movement model Creator Framework (MMCF) with states representing the user’s location. Using MMCF an optimal number of states $O(M|L,R,Q)$ Markov model can be determined for every $N(A,D,B)$ network and for $\varepsilon$ error vector $\{J1,J2\}$.

The idea behind MMCS is to determine a Markov movement model with minimal number of states for a given network that meets the initial conditions. These conditions are determined by the fault vector $\varepsilon$. In the model creator framework I have created recommendations for determining the parameters $L, R, O$ which are explained in the next subsections.

### 3.3.1 Error vector

The error vector is an input for the MMCF. The elements of the vector are the followings:
- $\varepsilon_{ds}$ the limit of mean difference between the transition matrixes for different user states.
- $\varepsilon_{rd}$ the acceptable error rate of outgoing prediction probability, when MAPs are merging together.
- $\varepsilon_{op}$ the limit of the difference of the outgoing probabilities of the sequences from current MAP.
- $\varepsilon_{uv}$ the limit of fluctuation of the number of users arriving from a certain sequence directions is investigated.

More details about the error rates in the determination of attribute sections.

### 3.3.2 Main type of the model

The proper main type of the model is determined by the goals and requirements, not by mathematical computation. Beside the mentioned examples there are some guidelines for selecting the best model type according terms and conditions:

- User-Centralized
  - Modeling from user point of view
  - User profile creation
  - Fraud detection
- Access Point-Centralized
  - Modeling from cell point of view
  - CAC in a MAP
  - Movement modeling of a geographical area
3.3.3 Determining the level

A new level should be applied in the model, if the mean difference between transition matrices for different user statuses is greater than a predefined limit. Of course if there is a requirement to use levels, then it must applied independently from the calculation.

Let us define $e_{ds}$ as limit of mean difference between the $B_{\Pi}$ and transition matrix for different $s$ user statuses ($B'_{\Pi}$). The average weighted deviation can be calculated by the following:

$$w'_{ds} = \frac{b \cdot abs(B_{\Pi} - B'_{\Pi})}{n_s^2}, \quad s \in S.$$  \hspace{1cm} (25)

If $w'_{ds} > e_{ds}$ is true for a status $s$, then a new level must be introduced into the model for status $s$. This inequality must be analyzed for all $s \in S$.

The number of levels can be calculated:

$$n_L = \sum_{\forall s \in S} \text{sign}(v'_{ds} + abs(v'_{ds})), \quad v'_{ds} = w'_{ds} - e_{ds}.$$  \hspace{1cm} (26)

3.3.4 Determining the resolution

As I explained earlier if outgoing predictions of users in more adjacent MAPs matches within a certain limit, then the MAPs could be merged together to create a new, major MAP. With this step the complexity of the model is decreasing.

I present a simple algorithms to determine the resolution. The input parameters of the algorithms are:

- $e_{rd}$, the acceptable error rate of outgoing prediction probability, when MAPs are merging together
- transition matrix, $B_{\Pi}$, as it was defined earlier
where $S_m$ the set of MAPs in the model, $S_{a\,a}^\prime$ the set of neighbouring MAPs of $MAP_a$, $row(A,a)$ the a-th row of matrix $A$, $col(A,a)$ the a-th column of matrix $A$. This results a smaller transition matrix, $B'_{m'}$, which leads to a simpler model.
3.3.5 Determining the memory

The future movement of the users is highly influenced by the path they have taken in the past to reach the investigated point. Leaving this out of consideration would introduce large errors into the mobility model. However it is not always useful to look back into each direction or to look back in equal depth into each direction from every MAP.

The determination of **memory** or **depth** needs proper precaution. The **depth** exponentially increases the number of states in the model. This can be seen in Figure 25, where the **depth** is generally 2, for all MAPs \(Q = [2, 2, 2, 2]\).

Main idea is to analyze the importance of each MAP sequence, visited by the users and decide its importance for consideration. The analysis starts with a sequence of length 2 (Length 1 means that only the current MAP is observed) and it is increased one by one. If a sequence of length \(i\) belongs to a MAP that is not important, then it will decreased, and \(i-1\) depth will denoted for the MAP. The importance of \(k\) depth is decided based on the following basic criteria:

- Take the MAP ID sequences for \(k\) length, which differ in the first MAP ID and belong to a current MAP. The difference of the outgoing probabilities of the sequences from current MAP must be investigated. Let us define \(\varepsilon_{op}\) as a limit for this difference. The difference for a MAP, and \(k\)-depth is determined the following way:

\[
Df[op]_k = \sum_{b \in Q^k} \sum_{a \in \Omega^k} \frac{\sum |b_{(a),l} - b_{(b),l}|}{n'_{am} (w_{vn}^k)^2}, \quad (27)
\]

where \(Q_a\) is the set of existing \(k\) length sequences from MAP, \(NB_i\) is the set of neighbor MAP IDs of MAP, and \(q\) denotes a sequence from the set.

The first criterion of importance is: \(Df[op]_{>\varepsilon_{op}}\).

- Take the MAP ID sequences for \(m\) length, which differ in the first MAP ID and belong to a current MAP. The fluctuation of the number of users arriving from a certain sequence directions is investigated. Let us define \(\varepsilon_{uv}\) as a limit for this variance. The variance of number of incoming users from a sequence into MAP:

\[
V^u_i = E \left[ E \left( \frac{n'^u_a}{n^u_a} - \frac{n'^a_u}{n^a_u} \right) \right] = \sigma \left( \frac{n'^u_a}{n^u_a} \right), \quad (28)
\]

where \(n'^u_a\) is the number of users in MAP arrived from a sequence (path), \(n^u_a\) is the number of users in MAP.

This must be examined for all of the incoming sequences:

\[
Df[uv]_k = \sum_{a \in \Omega^k} V^u_i \sigma, \quad (29)
\]

The second criterion of importance is: \(Df[uv]_{>\varepsilon_{uv}}\).

This two criteria, \(Df[op]_{>\varepsilon_{op}}\) and \(Df[uv]_{>\varepsilon_{uv}}\) must be applied for all MAP in order to determine \(Q\).

3.3.6 Optimal number of states with MMCF

The complexity of the given model could be denoted with the number of states represented by Equation 23, where \(n_m\) is the number of MAPs in the examined network, \(n'_{nl}\) is the number of new MAPs after merging for the resolution \(l\), \(w_{vo}\) is the weighted average depth on level \(l\), and \(w_{vn}^k\) is the average number of neighboring MAs on level \(l\).
The main point of the proof is that the attributes are defined by functions \( L = f(A, D, B, \varepsilon) \), \( R = f(A, D, B, \varepsilon) \) and \( O = f(A, D, B, \varepsilon) \) and the number of states \( n_{\text{states}} = f_{\text{states}}(L, R, O) \) can be derived from these. Function \( f_{\text{states}} \) indirectly depend on the error vector by \( f_l, f_r \) and \( f_o \) and as a function of this the number of states is strictly monotonically decreasing. Hence the proof is indirect, if there would be a Markov model for parameters \( A, D, B \) with a lower number of states, the corresponding \( \varepsilon^* \) would exceed the error vector \( \varepsilon \) given as the original condition (\( \varepsilon^* > \varepsilon \)).

With this step I have created the possibility for creating arbitrary discrete time Markov models. Below I have investigated the movement models that can be created by the MMCF.
3.4 Markov Models – Generalized, Mn model

In this section I describe some of the previous introduced, classified models.

3.4.1 M3 model

In the M3, Markov-chain based model [27] a user can be located in four different states during each time slot, the stay state (S), the left-area state (L), the right-area state (R) and outside-area (O). This classification of cells can be seen in Figure 27.

![Figure 27. Cells separated into three groups](image)

The grouping can be derived from the user behavior. If the users in right-hand side cells behave similarly from the current cell’s point of view, the neighboring cells will be merged into a common cell group, which represents a state in the Markov model (R state). Other grouping methods can be used as well, i.e. a standalone cell can constitutes a group also. In this example model each of the two groups (R and L) contains three cells. The state O represents the outside area, where users can come in to the L, and R state from, and where users can go from the L, and R state to.

Let us define $X(t)$ random variable, which represents the movement state of a given terminal during time slot $t$. I assume that $\{X(t), t=0,1,2,...\}$ is a Markov chain with transition probabilities $p$, $q$, $v$.

If the user is in state S of Markov model for cell i (current cell), it remains in the given cell. If the user is in state R, it is in range of the cells on the right-hand side, if in state L, it is in the left-hand side of the dividing line and if in state O, it is in one of the outsider cells.

Since the transition properties are not symmetric, the 'left area' state and the 'right area' state have different probabilities. Figure 19.a depicts the Markov chain and Figure 28 the transition (Π) matrix.

![Figure 28. The Π matrix of 3-state M-model(M3)](image)

Transition probabilities $p$, $q$ and $v$ can be determined based on the network parameters and are time dependent and different for each cell.

- $p_R$ – the probability of moving from stationary state (S) to right area state (R)
- $p_L$ – the probability of moving from stationary state (S) to left area state (L)
- $v_{LR}$ – the probability of moving from left area state (L) to right area state (R)
- $v_{RL}$ – the probability of moving from right area state (R) to left area state (L)
- $v_{LO}$ – the probability of moving from left area state (L) to outside area state (O)
- $v_{RO}$ – the probability of moving from right area state (R) to outside area state (O)
$v_{O,L}$ – the probability of moving from outside area state ($O$) to left area state ($L$)

$v_{O,R}$ – the probability of moving from outside area state ($O$) to right area state ($R$)

$q_R$ – the probability of staying in right area state ($R$)

$q_L$ – the probability of staying in left area state ($L$)

$q_O$ – the probability of staying in outside area state ($O$)

The balance equations for this Markov chain are given in Eq. (30):

$$
\begin{align*}
P_S \cdot (p_R + p_L) &= P_S \cdot (1 - q_R - v_{RL} - v_{RO}) + P_L \cdot (1 - q_L - v_{LR} - v_{LO}) \\
P_R \cdot (1 - q_R) &= P_S \cdot p_R + P_L \cdot v_{LR} + P_O \cdot v_{OR} \\
P_L \cdot (1 - q_L) &= P_S \cdot p_L + P_R \cdot v_{RL} + P_O \cdot v_{OL} \\
P_O \cdot (1 - q_O) &= P_R \cdot v_{RO} + P_L \cdot v_{LO}
\end{align*}
$$

The steady state probabilities, $P_S, P_L, P_R, P_O$ can be calculated.

With knowledge of the result the distribution of the users can be given in the environment of the current cell in a steady state. If we want to predict the number of mobile terminals for timeslot $t+1$ for each cell, using Eq. (31.a), where $N_i^t$ is the number of users in MAP$_i$ for timeslot $t$, $N_L^t$ and $N_R^t$ the number of users in left area and right area for timeslot $t$, and $p(i), v(i), q(i)$ means the transition parameters of the $i$-model:

$$
N_i^{t+1} = N_i^t \cdot (1 - p_R(i) - p_L(i)) + \\
+ N_L^t \cdot (1 - q_R(i) - v_{RL}(i) - v_{RO}(i)) \\
+ N_R^t \cdot (1 - q_L(i) - v_{LR}(i) - v_{LO}(i))
$$

(a)

$$
N_R^{t+1} = N_R^t \cdot q_R(i) + N_L^t \cdot p_R(i) + N_L^t \cdot v_{LR}(i) + N_O^t \cdot v_{OR}(i) \\
N_L^{t+1} = N_L^t \cdot q_L(i) + N_R^t \cdot p_L(i) + N_R^t \cdot v_{RL}(i) + N_O^t \cdot v_{OL}(i)
$$

(b)

It is possible to calculate the number of terminals for right area and left area side as well, in order to determine user distribution of neighboring cells (Eq. 31.b).

This model performs well when the user’s distribution in the left- or right-area state is uniformly.

The M3 model could be classed as an MMCF model, with the following parameters: $n_L=1, Q=[1,1,1], R=[1,\{2,3,4\},\{5,6,7\}]$. 

65
3.4.2 Generalized, Mn model

Thesis 2.3 I have created the n+2 state Mn model, which is an enhancement of the access point-centralized M3. The Mn mobility model is more accurate than the 3-state Markov model (M3). The accuracy difference in percentage is:

$$\sum_{k=L,R}^N \left( E \left[ \frac{N_i}{N_k} \right] N_k \cdot \lambda \right) \%,$$

where L, R are the right and left area MAP groups in M3 model, Ns the number of users in MAPs, $S^{(i)}_{adj}$ is the index set of neighbour MAPs of MAP, and $\lambda$ is the average handover rate.

The calculation complexity of the model is $o(n^3 + n + 1/n)$ [J3,J5,C2,C4,C5,C6].

If we try to predict the user’s distribution in a city having irregular, dense road system, or in a big park where people are able to move around then the handover intensities could differ thus the M3 model could produce errors. From this point of view the best way is if we represent all of the neighbor cells as a separated Markov state.

As I described above, in the unstructured models a MAP is simply mapped into a Markov-chain state, so 8 states are created, that is the 6 neighboring MAPs and the central MAP, plus all MAPs outside the cluster are represented by an ‘outside’ state. This results the M7 model [C4].

- stationary state ($S$)
- neighbour 1..6 state ($M_{N1}...M_{N6}$)
- outside area state

$$\Pi = \begin{bmatrix}
1 - \sum_{n=0}^6 p_n & p_1 & p_2 & p_3 & p_4 & p_5 & p_6 & 0 \\
1 - q_1 - v_{1,6} - v_{1,2} - v_{1,0} & q_1 & v_{1,2} & 0 & 0 & v_{1,6} & v_{1,0} \\
1 - q_2 - v_{2,3} - v_{2,1} & q_2 & v_{2,3} & 0 & 0 & 0 & v_{2,3} \\
1 - q_3 - v_{3,4} - v_{3,2} & v_{3,2} & q_3 & v_{3,4} & 0 & 0 & v_{3,2} \\
1 - q_4 - v_{4,5} - v_{4,3} & v_{4,3} & q_4 & v_{4,5} & 0 & 0 & v_{4,3} \\
1 - q_5 - v_{5,6} - v_{5,4} & v_{5,4} & q_5 & v_{5,6} & v_{5,4} & v_{5,6} & v_{5,4} \\
1 - q_6 - v_{6,5} - v_{6,1} & v_{6,1} & q_6 & v_{6,5} & v_{6,1} & v_{6,5} & v_{6,1} \\
0 & v_{6,0} & v_{6,0} & v_{6,0} & v_{6,0} & v_{6,0} & v_{6,0}
\end{bmatrix}$$

Figure 29: State diagram and Π matrix of eight-state Markov model

The Markov chain (Figure 20) and transition matrix (Figure 29) are more complex as in M3 case. Transition probabilities $p,q$ and $v$ have very similar determination:

- $p_k$ – the probability of moving from stationary state ($S$) to k. area state ($M_{nk}$)
- $v_{k,j}$ – the probability of moving from k. area state ($M_{nk}$) to right area state ($M_{nj}$)
- $q_k$ – the probability of staying in right area state ($M_{nk}$)

As in the previous case the steady state probabilities ($P_S, P_{N1}...P_{N6}, P_{N0}$) can be calculated. To calculate the number of users in MAP, for time slot $t$, then Eq. (32) is to be used:

66
\[ N_i^{t+1} = N_i^t \cdot \left(1 - \sum_{n=1}^{6} p_n(i)\right) + \]
\[ + \sum_{j, \text{MAP}_j \in S_{\text{adj}}^i} N_j^t \cdot \left(1 - q_j(i) - \nu_{j,j+1 \text{mod} 6}(i) - \nu_{j,j-1 \text{mod} 6}(i) - \nu_{j,o}(i)\right) \quad (a) \]
\[ \ldots \]
\[ N_k^{t+1} = N_k^t \cdot q_k(i) + N_k^t \cdot p_k(i) + N_{k-1 \text{mod} 6}^t \cdot \nu_{k-1 \text{mod} 6,k}(i) + \]
\[ + N_{k+1 \text{mod} 6}^t \cdot \nu_{k+1 \text{mod} 6,k}(i) + N_O^t \cdot \nu_{o,k}(i) \quad k \in S_{\text{adj}}^i \quad (b) \]

where \( N_i^t \) is the number of users in \( \text{MAP}_i \) for timeslot \( t \), \( p(i) \), \( v(i) \), \( q(i) \) means the transition parameters of the \( i \) model for timeslot \( t \), \( S_{\text{adj}}^i \) is the index set of neighbour \( \text{MAPs} \) of \( \text{MAP}_i \) and \( k \) is one of the neighbour \( \text{MAP} \) indexes.

A neighbour \( \text{MAP} \) index is the neighbour \( \text{MAP} \) direction from the center \( \text{MAP} \) point of view according the Figure 31 model. For example \( \text{MAP}_{31} \) is a neighbour of \( \text{MAP}_{17} \), and according the Figure 31 model it is in direction 5, then the neighbour \( \text{MAP} \) index is 5.

The M7 model could be classed as an MMCF model as well. The parameters: \( n_e=1 \), \( Q=[1,1,1,1,1,1,1] \), \( R=[1,2,3,4,5,6,7] \).

It is to be taken into account that in the real networks a cell does not always have six neighbours depending on the coverage. This model has to be generalized for a common case when a cell has \( n \) neighbour cells. I expanded my previous mentioned model to \( n \)-neighbour case (Figure 30), when all the \( n \) neighbours are represented with a Markov state:

- stationary state \( (S) \)
- neighbour 1..\( n \) state \( (M_{\text{N}_1}..M_{\text{N}_n}) \)
- outside area state
The steady state probabilities can be calculated as in the previous case:

\[
\Pi = \begin{bmatrix}
1 - \sum_{n=1}^{\infty} p_n & p_1 & p_2 & \cdots & \cdots & p_n & 0 \\
1 - q_i - v_{i,2} - v_{i,6} - v_{i,0} & q_i & v_{i,2} & 0 & \cdots & v_{i,n} & v_{i,0} \\
& \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\
& & \cdots & \ddots & \cdots & \cdots & \ddots \\
1 - q_i - v_{i,j+1} - v_{i,0} & 0 & \cdots & q_i & v_{i,j+1} & \cdots & v_{i,0} \\
& \ddots & \ddots & \cdots & \ddots & \ddots & \ddots \\
1 - q_i - v_{n,i} - v_{n,i+1} - v_{n,0} & v_{n,i} & \cdots & 0 & \cdots & q_n & v_{n,0} \\
0 & v_{0,1} & v_{0,2} & \cdots & v_{0,i+1} & v_{0,N} & q_0
\end{bmatrix}
\]

\(P_n \cdot P_k = \sum_{j=1}^{n} (1 - q_j - v_{j,j+1}\text{mod}\ n - v_{j,j-1}\text{mod}\ n - v_{j,0}) P_{Nj}
\]

\[P_N (1 - q_k) = P_S \cdot P_k + P_{N(k+1)} \cdot V_{k+1}\text{mod}\ n, k + P_{N(k-1)} \cdot V_{k-1}\text{mod}\ n, k + P_{NO} \cdot V_{i,0} \quad 1 \leq k \leq n \quad (33)
\]

\[P_{NO} (1 - q_O) = \sum_{j=1}^{n} P_{Ni} \cdot V_{i,0}
\]

Using the result the distribution of the mobile users is determinable in the steady state.

The predicted number of users in the next time slot is given in:

\[N_i^{t+1} = N_i^t \cdot \left(1 - \sum_{m=1}^{\infty} p_m(i)\right) + \sum_{j,MAP_j \in S_{adj}^i} N_j^t \cdot \left(1 - q_j(i) - v_{j,j+1}\text{mod}\ n(i) - v_{j,j-1}\text{mod}\ n(i) - v_{j,0}(i)\right) \]

\[\cdots (a)
\]

\[N_k^{t+1} = N_k^t \cdot q_k(i) + N_i^t \cdot p_k(i) + \sum_{j,MAP_j \in (S_{adj} - S_{adj})} N_j^t \cdot v_{j,k}(i) + N_O^t \cdot v_{O,k}(i) \quad k \in S_{adj}^i \quad (b)
\]

Figure 30: State diagram and \(\Pi\) matrix of \(n+2\)-state Markov model (\(M_n\))
Because all of the neighbor MAPs is represented as separated states, the prediction based
Mn model is more accurate than M3 model, or at least as accurate as M3 model.

The inaccuracy of prediction $N_i^{t+1}$ from MAP $j$ is the following:
\[
\left(1 - \frac{N_i'}{N_k'}\right)N_k' \cdot b_{j,i}, \tag{35}
\]

where $N_k'$ is the number of users in the group (left-area or right-area in M3 model)
where MAP $j$ belongs.

This is calculated for all MAP and weighted with the number of users in the group:
\[
\sum_{K=L,R} \sum_{i \in S_k} \left\{\frac{1}{3} \left(1 - \frac{N_i'}{N_k'}\right)N_k' \cdot b_{j,i} \right\}. \tag{36}
\]

Let me determine the inaccuracy in other way. M3 compared to Mn by weighting the two
models prediction (Eq. 31a and Eq. 34a) difference for the next moment with the number of
actual users:
\[
\frac{N_i^{t+1} - N_i^{t+1}_{M3}}{N_i^{t+1}_{Mn}}. \tag{37}
\]

Instead of transition probabilities $p, q, v$ the elements of the rate matrix $B_0 = [b_{ij}]$ are
used. The Eq. 38 shows the prove, which follows the same result as in Eq. 36:
\[
\left(\sum_{j \in S_{adj}(L)} N_j \cdot b_{L,i} + \sum_{j \in S_{adj}(R)} N_j \cdot b_{R,i} \right) - \left(\sum_{j \in S_{adj}(K)} N_j \cdot b_{j,i} \right) =
\sum_{K=L,R} \sum_{j \in S_{adj}(K)} \left(\sum_{i \in S_{adj}} N_j \cdot b_{j,i} \right) - \left(\sum_{j \in S_{adj}} N_j \cdot b_{j,i} \right) =
\sum_{K=L,R} \sum_{j \in S_{adj}(K)} \left(\frac{N_j}{3} - N_j \right) \cdot b_{j,i} =
\sum_{K=L,R} \left(\frac{1}{3} N_j \cdot b_{j,i} \right) =
\]

To generalize the equation instead of $b_{ji}$ the average handover rate, $\lambda$ is used:
\[
\sum_{K=L,R} \sum_{j \in S_{adj}(K)} \left\{E \left[1 - \frac{N_j}{N_k}\right] \cdot N_k' \cdot b_{j,i} \right\}. \tag{39}
\]
In M3 model the neighbouring MAPs are grouped into two areas. But it is possible any other grouping as well. For these cases equation 33 transformed into fully general form:

\[
\sum_{j, MAP \in S_j} \left\{ \sum_{k, NG} \left[ \frac{1}{N_G + 1} \left( \frac{N_j}{N_K} \right) N_K \cdot \hat{\lambda} \right] \right\},
\]

(40)

where \( G \) is the set of areas, and \( n_G \) is the number of areas in the model.

I calculated the calculation complexity also as the function of state number. The calculation complexity is determined by three main factors: processing capacity needed for the determination of the crossing probabilities, calculation of a steady state of the gauss elimination and the prediction.

Based on the \( n+2 \)-state model the prediction computation is calculated. Let me define a common, unit computation cost:

- \( c_{comp} \) – the unit cost of a mathematical operation, memory using, or file operation.

The computation cost consist three main steps:

- determination of the transition probabilities: \( n \cdot 3 \cdot c_{comp} + (n + 1) \cdot c_{comp} \)
- gauss elimination in order to get the steady state probabilities (Eq. 33) from transition matrix (Figure 30): \( (n + 2)^3 \cdot c_{comp} \)
- prediction (Eq. 28): \( 2 \cdot (n + 2) \cdot c_{comp} + \frac{n_M}{n + 2} \cdot c_{comp} \)

Summarized these:

\[
n \cdot 3 \cdot c_{comp} + (n + 1) \cdot c_{comp} + (n + 2)^3 \cdot c_{comp} + 2 \cdot (n + 2) \cdot c_{comp} + \frac{n_M}{n + 2} \cdot c_{comp} = \gamma \]

\[
= c_{comp} \cdot [(n + 2)^3 + 6n + 5 + \frac{n_M}{n + 2}] \quad .
\]

(41)

Based on Eq. 35 the complexity can be estimated with:

\[
o(n^3 + n + \frac{1}{n}) \quad .
\]

(42)
3.4.3 Simulation results

In this chapter I compare the accuracy of the mentioned Markov model to other models found in the literature, and validate my $Mn$ model. The simulation environment of a cell cluster shown in Figure 31.

![Figure 31: The logical cell-cluster in the simulation environment](image)

The simulation was written in the open source OMNet++ using C++ language (Figure 32). The simulation environment consisted of a cluster with 61 named cells and it also included geographical data that is interpreted as streets and a park on the cluster area. The drift of the movement is heading to the streets from neutral areas.

![Figure 32: The developed cell-cluster in OMNet++ simulation environment](image)

The simulation used 610 mobile terminals (10 for each cell), in the initial state uniformly distributed in the cluster. The average motion velocity of the users is parametrized with a simple phase-type (PH) cell dwell time simulator (reciprocal of exponentially distributed values). In the simulation time mobile terminals appear and disappear, in order to simulate the active and inactive states.
The simulation consists of two parts. The trace simulation is the series of cell-transitions that the mobiles have initiated. It produces a time-trace that contains the actual location data for each mobile terminal in the network (reference interval). I have used this trace simulation as if it was a provider’s real network trace.

The second part is the estimation procedure that uses the past and the current reference simulation results to estimate future number of users in each cell. The estimation error is interpreted as the measure of accuracy of each mobility model in this Dissertation.

The prediction starts 100 timeslots after the reference simulation initiation. During the warm-up process the reference simulation produces enough sample data for the correct estimation, which uses the previous reference results as an input to estimate the future user distribution. Each user-transition in the 100-timeslot reference period is used to derive transition probabilities, motion speed and patterns in the simulation cell-space. These patterns serve as an input for the simulation threads of each mobility model. The models have the same input throughout the simulation process so that the results are comparable.

A widely used modified Random Walk estimation, M3[27] models and extended Random Walk[27] were used in the simulation as references. I used the M7[26] model to validate my $Mn$ model because the M7 is a special case of the $Mn$. The extended Random Walk model provides the possibility of staying in the same cell for the next fixed length time slot as an extension of the RW model. There is an additional transition, the loopback direction. The simple RW model does not allow the user to stay in the same state, so my proposition enables the model to simulate different cell dwell times. Formally that means the probability of stepping into the same cell at the end of the slot.

In the first section the estimation errors of the models in the simulation are measured from two aspect. In both examination the average error of the user number estimations calculates in the MAPs in each time slot. It produces a time-dependent average error value (TAEV) in each timeslot for the cell-cluster. TAEV shows the average error compared to the actual user number in the cells.

In the first approach the TAEV is examined according to the dynamics of the motion. The generic L parameter effects the motion velocity of the simulated users. The higher value means shorter time, which the users spend in a MAP. The motion pattern of this case is close to the simple uniform motion pattern. The results are depicted in Figure 33.
The TAEV oscillates around the aggregated average error level (AAEL) of mobility models. AAEL is the average of the TVEL. Figure 34.a shows the AAEL of the models in every L. The
models estimations are improving with L increasing. Figure 34.b depicts the standard deviation of the TVEL (SDT).

In the second approach the TAEV is examined according to specific and uniform motion of the users. It is two basically different simulation environments.

Six different scenario is measured. In the first scenario the users always move into one specific direction from every cell, there is no practical chance of stepping out of the path. This case simulates for example users on a highway, or a road.

In the second case two specific direction is set, and so on until six direction. This last scenario represents a crowded urbanized area, where each direction has a nonzero probability. Figure 35 shows the TAEV with 1,2, and 4 specific direction settings. AAEL and SDT are plotted in Figure 36 with all number of directions.
Figure 35: The TAEV values in RW, ExtRW, M3 and M7 models with direction=(1,2,4)
One can see that the Random Walk based models and M3 as well are not able to follow the motion patterns, they work with a significantly higher error rate than the M7 model. As the number of specific direction increasing, the users motion pattern are getting closer to the simple uniform motion pattern the difference between the estimations are decreasing dramatically. Figure 36.a shows the characteristics of the increasing the estimation accuracy. The standard deviation of the models (Figure 36.b) does not show significantly correlation with the number of motion directions.

To sum up the simulation results the M7 model is by 12.67% more accurate than M3, and by 81.85% than RW model.

I have also investigated the computational requirement of the mobility models. The relative performance of the models is shown in Figure 37 on a logarithmic scale.
In the second section I used the MMCF parameter calculation algorithms, introduced above in chapter 3.4 for the simulation environment. The smaller examined area contains the cells in the red circle (cell 1-7, cell 16-18) in Figure 31. In MMCF generated optimal markov model estimation compared to the fix M3, M7 models. The input parameters of MMCF for this simulation environment:

\[ S = \{\text{handover during voice call}\}, \varepsilon_c = 5, \varepsilon_{uv} = 0.4, \varepsilon_{op} = 0.2, \]

I examined only the handover event, so the \( D \) matrix is empty and because of the limits of this Dissertation the \( C \) and \( B_Q \) matrices are not presented. Structured, access point centralized model with one level was chosen. The result of the algorithms:

\[ O = \{[2,1,1,2,1,2,1,1,2,1,1]\} - \text{which means for MAPs 1,6,17 the depth is 2, for the others it is 1,} \]

\[ R = \{1,2,3,4,5,6,7,16,17,18\} - \text{which means that MAP 4 and 5 is merged together} \]

and the Markov-chain is the following (for clear interpretation not all of the edges depicted):

![Markov-chain of the generated model by MMCF](image)

The following plots (Figure 39.) show the average error of the estimations in every \( t \) timeslot. Random Walk model performed worst, it cannot follow the patterns in user fluctuation as it was expected. The M3 and M7 models work with significantly lower error rate, but in \( t=105, t=120 \) and \( t=135 \) timeslots the average error rate increased suddenly. This is caused by the change of distribution of the directional moving users (suddenly increased the number of active mobile users), what the simple Markov models cannot follow. The MMCF generated Markovian approach holds the average error rate, it followed the changes in user motion appropriately, it is able to learn the directional motion patterns and the fluctuation of user distribution, which proves the strength of the Markov Model Creator Framework.
Figure 39: The TAEV values in RW, M3, M7 and optimal MMCF model
3.5 Determining the parameters and calculating the empirical error of the generalized, Mn model

In previous chapters only the theoretical error of the Markov models are handled. To be able to determine the users’ future distribution we need to deduce the steady state probabilities of the Markov chain and for this we need to define the transition probabilities. The transition probabilities can be defined as the relative frequencies calculated from the network logging. This process is not accurate, the determination contain errors depending on the number of samples. I have named this error as empirical error and denoted it with $E_p$.

This chapter handles the parameter determination of previous introduced Mn model, and the empirical error rate ($E_p$) of steady state probability calculation.

**Thesis 2.4** I have introduced a method to process the network traces and determined the transition probabilities of Mn model. Using this method I have determined the empirical error ($E_p$) of the user distribution determined by the Mn model:

$$H_{mn} = H_S + nH_{Hk} + H_{NO},$$

where

$$H_S = n(1-q-3v)H_{Hk} + (1-np)H_S - \frac{4}{\sqrt{2}}(nH_{Hk} + 1 + H_S),$$

$$H_{Hk} = H_S p + H_{Hk} (2v+q) + H_{NO} v + \frac{1}{\sqrt{2}}(H_S + 3H_{Hk} + H_{NO} + 1),$$

$$H_{NO} = nH_{Hk} v + H_{NO} q + \frac{n}{\sqrt{2}}(H_{Hk} - H_{NO}),$$

and $p,q,v$ are the averages of the transition probabilities of the Mn Markov model [J3,C2,C4].

Network operators log the events in their network, amongst them the users’ cell transitions between MAPs. I have developed a procedure for summing up these latter events. Using this based on relative frequency the transition probabilities of Mn model can be determined.

As it was described earlier in $P$ matrix a cell and a status is appended to every user in every timeslot. Let me determine an $s_{ij}$ event, $s_{ij} \in S$, which denote the step event into a MAP. From $P$ matrix reduced $P_{SH}$ matrix can be given, that the handover matrix $B$ can be approximated from.

First of all, the transition probabilities of Mn model, used in previous section, are defined as:

- $p_s$ = the probability of moving from stationary state ($S$) to $M_a$
- $q_s$ = the probability of staying in a move state ($M_a$)
- $v_{a,m}$ = the probability of moving from a move state ($M_a$) to another one ($M_a$)

The determination of transition probabilities are based on relative frequency. The relative frequency is not accurate. Accuracy of this calculation depends mainly on the number of samples. That is a finite set of samples could not be sufficient, there is always a minimal error which is the difference between the real, model ($p_i, q_i, v_{i,k}$) and the calculated ($\hat{p}_i, \hat{q}_i, \hat{v}_{i,k}$) values. Let me define the $e_{p,i}, e_{q,i}, e_{v,i,k}$ as the error of $p_i, q_i, v_{i,k}$ determination of the parameters of the Markov model (As $e_{p,i}$ in Eq. 37, $e_{q,i}, e_{v,i,k}$ can be determined similarly).

The mean value of calculated parameters (for example $E(\hat{p}_i)$ is the real value because of the law of averages: 79
As I defined earlier the n-state Markov model (see chapter 3.5.2) satisfies the general matrix equation \( P = P\Pi \), which can be solved. But I am not able to determine exactly the \( \Pi \) matrix. Calculating with the relative frequencies I get a \( \hat{\Pi} \) matrix, which estimates the \( \Pi \) matrix with calculation errors. When I solve the matrix equation \( P = P\hat{\Pi} \) using \( \hat{\Pi} \) instead of \( \Pi \), I get a \( \hat{P} \) equals \( P \) with the addition of the model error. Summarizing these coherences:

- theoretical solution : \( P = P\Pi \),
- practical solution : \( \hat{P} = \hat{P}\hat{\Pi} \),

\[
\hat{P} = P + H,
\]

\[
\hat{P} = \hat{P}\hat{\Pi} \rightarrow P + H = (P + H)(\Pi + E)
\]

\[
H = PE + H\Pi + HE
\]

where \( H \) vector denotes the error of the steady state probability coming from the parameter calculation error, and \( E \) means the matrix derived from the \( \epsilon_p^i, \epsilon_q^i, \epsilon_v^i \). The sum of rows in \( E \) matrix must be equal to 0, because the \( \hat{\Pi} \) matrix which is \( \Pi + E \), is also a transition matrix, and the sum of rows must equal to 1.

\[
E = \begin{bmatrix}
-\sum_{i=1}^{n} \epsilon_p^i & \epsilon_p^1 & \epsilon_p^2 & \ldots & \epsilon_p^n & 0 \\
-(\epsilon_q^1 + \epsilon_q^{1,2} + \epsilon_q^{1,3} + \epsilon_q^{1,0}) & \epsilon_q^1 & \epsilon_q^2 & \ldots & \epsilon_q^n & 0 & \epsilon_v^1 & \epsilon_v^{1,0} \\
& \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
-(\epsilon_q^{j-1} + \epsilon_q^{j+1} + \epsilon_q^{j,0}) & \epsilon_q^{j-1} & \epsilon_q^{j} & \epsilon_q^{j+1} & \ldots & \epsilon_q^{n} & 0 & \epsilon_v^{1} & \epsilon_v^{j,0} \\
& \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
& 0 & \epsilon_v^{0,1} & \epsilon_v^{0,2} & \ldots & \epsilon_v^{0,n} & \epsilon_v^0
\end{bmatrix}
\]

Figure 40: The \( E \) calculation error matrix

Let me do some simplifying by calculating average of error parameters:

\[
\epsilon_p = \text{avg}(\epsilon_p^i) \quad \forall i
\]

\[
\epsilon_q = \text{avg}(\epsilon_q^i) \quad \forall i
\]

\[
\epsilon_v = \text{avg}(\epsilon_v^{i,j}) \quad \forall i, \forall j
\]

where \( \text{avg} \) is the average number. After this I got the following \( E \) matrix, which is used the next steps:
My aim by the following is to calculate the dependency of the model error on the network parameters. Let me start from the Eq. 31. described with the calculated parameters. Replace the variables by Eq. 31 and 45 to get the following equation (for simplification further the \( k+1, k-1, j+1, j-1 \) indexies are always interpreted with mod \( n \)):

\[
\begin{align*}
(P_S + H_S) \cdot \sum_{i=1}^{n} (p_i + \epsilon_p) &= \sum_{i=1}^{n} \left( 1 - (q_i + \epsilon_q) - (v_{j,i-1} + \epsilon_v) - (v_{j,i} + \epsilon_v) - (v_{j,i+1} + \epsilon_v) \right) (P_{N_i} + H_{N_i}) \\
& \quad \cdots \\
(P_{N_i} + H_{N_i}) (1 - (q_i + \epsilon_q)) &= (P_S + H_S) \cdot (P_{N_i} + \epsilon_p) + (P_{N_{(k+1)}} + H_{N_{(k+1)}}) \cdot (v_{k+1} + \epsilon_v) \\
& \quad + (P_{N_{(k-1)}} + H_{N_{(k-1)}}) \cdot (v_{k-1} + \epsilon_v) + (P_{N_O} + H_{N_O}) \cdot (v_{i,O} + \epsilon_v) \quad 1 \leq k \leq n \\
& \quad \cdots \\
(P_{N_O} + H_{N_O}) (1 - (q_O + \epsilon_q)) &= \sum_{i=1}^{n} (P_{N_i} + H_{N_i}) \cdot (v_{i,O} + \epsilon_v)
\end{align*}
\]

As next step I denote \( H_S, H_{N_i} \) for every \( i \) and \( H_{N_O} \) using the terms from Eq. 31., Eq. 45 and the coherence \( \epsilon = \max(\epsilon_p, \epsilon_q, \epsilon_v) \). In addition let me define a \( H_{N_k} = \max(H_{N_i}, \forall i) \) simplifying:

\[
H_S = [\sum_{j=1}^{n} (1 - q_j - v_{j,i+1} - v_{j,i} - v_{j,i-1}) H_{N_k} + (1 - \sum_{j=1}^{n} p_j) H_S] + \\
[-n\epsilon(P_S + H_S) - \sum_{j=1}^{n} 4\epsilon(P_{N_i} + H_{N_k})]
\]

\[
H_{N_k} = [H_S p_k + H_{N_k} v_{k+1} + H_{N_k} v_{k-1} + H_{N_k} q_k + H_{N_O} v_{i,O}] + \\
[\epsilon(P_S + H_S) + \epsilon((P_{N_m} + H_{N_k}) + (P_{N_{(m-1)}} + H_{N_k}) + (P_{N_{(m+1)}} + H_{N_k}) + (P_{N_O} + H_{N_O})] \quad 1 \leq m \leq n \\
\]

\[
H_{N_O} = [\sum_{j=1}^{n} H_{N_k} v_{j,i} + H_{N_O} q_O] + [\sum_{i=1}^{n} \epsilon H_{N_k} - n\epsilon H_{N_O}]
\]

\[
\sum_{k=1}^{n} P_{N_k} = 1 - P_S - P_{N_O} \Rightarrow \sum_{k=1}^{n} P_{N_k} \leq 1 - P_S, \quad (49)
\]

\[
P_{N_m} + P_{N_{(m-1)}} + P_{N_{(m+1)}} + P_{N_O} \leq \sum_{k=1}^{n} P_{N_k} + P_{N_O}. \quad (50)
\]

Using the Eq.49 and Eq.50 upper estimations the \( P \) vector eliminated:
Let me do some simplifying again by calculating average of some parameters in following way:

\[
p = \text{avg}(p_i) \quad \forall i
\]

\[
q = \text{avg}(q_i) \quad \forall i
\]

\[
v = \text{avg}(v_{i,j}) \quad \forall i, \forall j
\]  \hspace{1cm} (52)

These three general probability parameters can describe typical user movements in the current cell. For example \( p \) means the probability of the moving after, \( q \) is for describing how the user can hold its current position in the examined cell. Using these general parameters yields an upper estimation with an equation system in 2 variables:

\[
H_S = n(1-q -3v)H_{S_k} + (1- np)H_S - 4\varepsilon(nH_{S_k} + 1 + H_S)
\]

\[
H_{S_k} = H_S p + H_{S_k}(v_k+q_k + H_{S_O}) + (H_S + nH_{S_k} + H_{S_O})+ 1\varepsilon
\]  \hspace{1cm} (53)

\[
H_{S_O} = nH_{S_k} + H_{S_O}q + n\varepsilon(H_{S_k} - H_{S_O})
\]

If I examine the \( \varepsilon \) error rate, I find that an upper estimation can be applied for it as well using Chebyshev inequality. The variance of relative frequency is well known:

\[
\sigma(\hat{p}) = \sqrt{\frac{p(1-p)}{m}}, \hspace{1cm} (54)
\]

where \( p = E(\hat{p}_i) \), and \( m \) are the number of samples in the trace. Let me make an upper estimation on Eq. 48 to eliminate variable \( p \):

\[
\sqrt{\frac{p(1-p)}{m}} \leq \sqrt{\frac{1}{4m}} = \frac{1}{2\sqrt{m}} \hspace{1cm} (55)
\]

Using this result in Chebyshev inequality:

\[
P(|\hat{p} - E(\hat{p})| \geq \varepsilon) \leq \frac{1}{2m\varepsilon^2} \hspace{1cm} (56)
\]

If I assume that \(|\hat{p} - E(\hat{p})|\) is not larger than \( \varepsilon \) with 99\% probability, and the \( m=10000 \), I reach an upper estimation for \( \varepsilon \):

\[
0.01 \leq \frac{1}{2\sqrt{m\varepsilon^2}}
\]

\[
\ldots \hspace{1cm} (57)
\]

\[
- \frac{1}{\sqrt{2}} \leq \varepsilon \leq \frac{1}{\sqrt{2}} \rightarrow \varepsilon = \frac{1}{\sqrt{2}}
\]

In order to get an equation system depends only on \( p, q, v \), the \( \varepsilon \) is replaced in Eq. 53:
\begin{align*}
H_S &= n(1-q-3v)H_{Nk} + (1-np)H_S - \frac{4}{\sqrt{2}}(nH_{Nk} +1 + H_S) \\
H_{Nk} &= H_S p_m + H_{Nk} (2v+q) + H_{NO} v + \frac{1}{\sqrt{2}}(H_S + 3H_{Nk} + H_{NO} + 1) , \quad (58) \\
H_{NO} &= nH_{Nk} v + H_{NO} q + \frac{n}{\sqrt{2}}(H_{Nk} - H_{NO})
\end{align*}

\(H_S, H_{Nk}, H_{NO}\), the error of the steady state probability can be calculated from the equation system. As last step I define \(\text{H}_{\text{Mn}}\), the model error from \(H_S, H_{Nk}, H_{NO}\):
\[H_{\text{Mn}} = H_S + nH_{Nk} + H_{NO}. \quad (59)\]

In Eq. 59, I calculated the error of the Mn model. Using the plotting abilities of Mathematica [P1] I examined this model error deepening on \(p,q,v,n\). The aggregated transition probabilities \(p,q,v\) describe the basic characteristics of the user movements, I have analyzed some characteristics of these. \(p\) always describe movement from the central cell toward a neighboring cell, \(q\) mean staying in the cell, \(v\) corresponds to transitions between neighboring cells. Using these the empirical error of \(Mn\) models prediction can be determined for different user movement patterns and behaviors.

I analyzed some different scenarios considered as typical user movements. The result of the first analyzing is depicted on Fig. 42. In this case the empirical error with the vary of the states number is examined. Three different cases are shown:
\begin{enumerate}
\item[a.] \(p\) and \(v\) transition parameters take low, \(q\) takes high value, meaning that the users in the current cell move slowly, stop several times.
\item[b.] all of the general transition parameters \((p,q,v)\) take low value. This means that the users head to the central cells.
\item[c.] this third scenario shows the opposite of the a. previous case, namely fast movement in all direction, none of the users stops.
\end{enumerate}
The characteristic of the diagrams is very similar, decreasing to a specific number of states, where the empirical error is 0, and after that they are increasing. The differences are the zero-beats and the gradient of the diagrams.

One can see that the higher the motion speed is, the higher the gradient is and the the lower the number of states where zero-beat exists.
Figure 42: Empirical error with the vary of the states number. a: $p=0.1$, $q=0.9$, $v=0.1$ (slow motion), b: $p=0.1$, $q=0.1$, $v=0.1$ (heading to the central map), c: $p=0.9$, $q=0.1$, $v=0.9$ (fast motion)

The second figure (Fig. 43) shows the empirical error with the vary of the $q$ transition probability. Two different scenario is examined, fast and slow motion. The same can be concluded like in previous case, with fast motion the empirical error is higher.

Figure 43: Empirical error with the vary of the $q$. a: $p=0.2$, $v=0.1$, $n=15$ (slow motion), b: $p=0.9$, $v=0.2$, $n=15$ (fast motion)

The last figure (Figure 44.) summarize the results, if the number of states higher, and the motion is faster, then the empirical error is increasing.
One can see in figure 42 and 44 that the empirical error decreases until a certain point and in all cases will increase by increasing the number of states. That this increase occurs at a higher number of states in case of slow motion and a lower number of states in case of fast motion and their steepness is different can be explained by the fact that lower in case of slow motion the number of different movement patterns is lower, hence the calculation error is lower. In case of fast motion the number of different samples is significantly higher. It can be clearly seen as well that in an situation a point is given where the empirical error is 0. I have analyzed by what $p$, $q$ and $v$ aggregated transition possibilities does this ideal situation exist. Figure 45 shows the eliminated empirical error with steady $v$. One can see by increasing the number of states the intensity of movement from the central cell toward a neighboring cell ($p$) is decreasing. The $p$ value is almost independent from $q$, by higher value of $q$ $p$ is little bit higher.

Figure 44. Practical error with the vary of the $p$ transition probability and number of states. a: $q=0.9$, $v=0.3$ (slow motion), b: $q=0.1$, $v=0.9$ (fast motion)

Figure 45. Eliminated empirical error ($E_P=0$) as a function of $p$, $n$, and $q$, $v=0.02$. 
In this section I determined the previously introduced Markov model parameters from the network traces, defined that the model empirical error derived from the parameter calculations, and examined it dependently on the user movement behaviors.
4 Client-driven Mobility Frame System – Mobility from a brand new point of view

I have compared the different position management strategies using more aspects during the creation of the mobility management model (MMM, Chapter 2.1). It can be simply seen that different mobility management strategy results the lowest total cost \((C)\) for certain networks. Typically they use a given strategy in a network, which supposedly results in optimal cost for the majority of the mobile users, but surely not for all. The ideal situation would be if all of the mobiles would follow such strategy that results in the lowest total network cost for each and therefore they would only use the minimally needed resources to realize their own position management.

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 or simple protocols (like Mobile IPv6 - MIPv6 [18]) by recognizing, executing and forwarding simple messages generated by the mobile 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. The network nodes provide only basic services for mobile entities: connectivity and administration.

I construct a framework called the Client-based Mobility Frame System (CMFS) for this mobility environment. I do not say that I have found the optimal system to provide IP or other kind of mobility, but I came 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. I show how to apply the classical strategies like the simple MIP, hierarchical, tracking or cellular approaches to my system and furthermore I propose an algorithm that creates cell structures efficiently and individually for each MN.

4.1 Client-driven Mobility Frame System (CMFS)

In this Section I will introduce my concept, the Client-driven Mobility Frame System (CMFS) in detail. I specify the basic roles of the network and what capabilities the fix 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. It is essential that the network nodes understand the commands from the mobile client that will be the brain and the director of a given mobility management system itself. Even each client can use different management strategies on the same network with the same technology and capabilities.

**Thesis 3.1**  
I have introduced a new, in mobile network used self-organizer location management framework, the CMFS (Client-driven Mobility Frame System). In CMFS every mobiles manage the mobility for itself and network provides only basic features, the connectivity and administration. Using CMFS every mobile or group of mobiles is able to select the mobility strategy which cause the minimum signaling cost from the network point of view \([J4,J6,J9,C3]\).
4.1.1 The idea

The client driven mobility management I introduce is inspired and based (but not depend!) upon the fact that a mobile 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 mobile node should always be able to have an up-to-date information of the nodes of this network. The size of this depends on the algorithm the mobile uses.

To give an example I will show later that to implement a basic MobileIP-like solution on my framework system the mobile only has to maintain information about 3 (or even only 2) nodes in the network, so far a node with very limited capacity this can be a solution good enough with a cost similar to the cost of Mobil IP. On the other hand, more complex entities like PDAs can maintain information about thousands of MAs and thus optimize communication costs or quality.

The most important advantage of my solution is that the service providers do not have to choose an exact mobility approach, which could be very inefficient in certain cases for certain users. 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. One more thing I 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 I believe that my solution has a better network resource utilization than any other classical one.

4.1.2 Basic Notations and Requirements

The basic notations will be the same as in my previous sections (3.2, 2.1).

To be able to serve the MNs and their algorithms I define requirements for the network. Once the MAs are aware of all these the 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. 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 does not matter whether the MN connected via WLAN or LTE or even MIPv4 or MIPv6 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 my CMFS will be implemented in the MN and are discussed later. The good thing in the requirements above is that the CMFS protocol is totally independent of all the underlying technologies no matter whether the communication goes over IP, ATM in a LTE, WSDMA environment or anything else; the important thing is that the nodes find each other. Note that not all the nodes in the network must necessarily have the CMFS implemented on, but only the ones will be recognized by MN.

The central MA (from Mobile IP it can be called Home Agent, 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 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.
4.1.3 The CMFS in the mobile node

We have seen what properties a network should have to be able to adopt my solution. Now I define the kind of logic that should be implemented into the network. I do it in the following way. At first I show how the MN can find out the network structure and then I give an explicit specification of the commands i.e. the messages the MN sends to the network nodes and the actions they take.

4.1.3.1 Network discovery – Logical Network

The first task for the MN is to discover the network. It can send a message to any of the MAs or MAPs it knows (the MN has to know at least one MA, the one that plays a Central Agent (for example Home Agent in Mobile IP)-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 IDs (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 builds a Logical Network (LN) it uses as an input for its algorithm (Figure 48).

![Figure 48: Build up a Logical Network](image)

The solution I have chosen in my 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 automatically generate an ICMP Time Exceeded message. The costs of using these links are determined by measuring the delay.

The traceroute is not always allowed at the nodes of the network providers. In this case a inner CMFS control message, CMFSTraceRoute has to be implemented, which would work similar to tracroute message introduced just now. 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 I think that it gives a better view of the real status of the network. For example if a node is down, it does not reply to the MN, i.e. the MN thinks that it does not exist, what is actually true from the MN’s point of view since the MN can communicate with operating nodes only.

4.1.3.2 Messages and actions

The main element of my solution is that the MN orders the MAs to modify a kind of routing database they maintain. The database in MA has entries like: "The MN can be reached via the MA, or MAP;" 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, the packet has a correct destination on the mobility level, only routing provided by the underlaying network is needed. 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 ordered by MN. MN can instruct an MA, when it attaches to a new MAP after a successful handover it naturally registers there. Let us construct such a register message structure:

```
[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]
 ]]
```

As you can see, these registration instruction are embedded into each other. As I mentioned above the action in the structure can be a Register, a Delete, or another instruction structure, which has to be forwarded by the actual MA. What the MA should do is to understand this message and maintain the entry in its database: if the paged node is MN then it should be searched via HA, MAn, ..., when MN is searched at MA, then MA knows that it can be reached via MAP1, MAPa, MAPa meaning that the packet is routed to all the 3 nodes representing a CIP-like algorithm [3]. If there is no such multiple route and the messages does not always contain the HA, then HMIP-like [33] is implemented. If there is an update that goes from MN directly to HA, then MIP-like [30] is implemented, if these last two kind of messages are mixed then a DHMIP-like [31] approach is presented. If the node sends messages like this:

```
[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−1]
]
```

then a HAWAII-like protocol [36] is implemented.

I have provided an example implementation 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.
4.2 The MN based management strategies

I have shortly introduced the new idea and explained the basics of its operation. In this section I will show possible strategies the MN can use to make the mobility management work. I 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 I am working in the university and I spend most of the day in my room and in two labs thus stay under a set of agents and access points. However, in the afternoon and in the morning I travel long distance passing through many MAPs. At home I have a router that is my HA. Intuitively I can think that the cellular approach is useful at the university and a tracking-like solution is the most efficient on the way home. Example mobility management strategies in CMFS are coming now.

4.2.1 Mobility management strategies using CMFS, inspired by classical solutions

Here I show how to implement the versions of the basic ip mobility protocols, such as hierarchical, cellular, Mobile IP etc. on the CMFS system.

4.2.1.1 Personal Mobile IP – PMIP

The operation of Personal Mobil IP is simple and easy. 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.

$$\begin{align*}
\text{[Dst: MAP}_{i}, \text{Src: MN, Actions: Register MN to MAP}_{i} \text{ via MN;}} \\
\quad \text{[Dst: HA, Src: MAP}_{i}, \text{Actions: Register MN to HA via MAP}_{i};} \\
\quad \text{[Dst: MAP}_{i-1}, \text{Src: MAP}_{i}, \text{Actions: Delete MN in MAP}_{i-1} \text{ via MN]} \\
\end{align*}$$

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. I would like to point out that the MN has to maintain a Logical Network of 3 node only. However, a great benefit of my proposal is that any MN can implement different version (e.g. soft handover) of the protocol without any modification in the network entities.
Figure 49: 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.

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 in Figure 49.

4.2.1.2 Personal Hierarchical Mobile IP – PHMIP

In the classical hierarchical Mobile IP there is a fixed network architecture that is a tree of nodes of Mobility Agents as Hierarchy Points (HP) and leaves of Mobility Access Points. The MN updates only to the nearest HP thus saving signaling load on the HP-HA route. The operation of a HMIP micro-mobility (talking about only two layered hierarchy) would give us the question: which node should be the MA in the hierarchical mobility approach. I suppose that seeing the traceroute messages, the MN can decide it. After that the command messages are again simple and easy to construct.

More problem arise when talking about multiple layered hierarchical solutions (Figure 50). 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 is easy again, since there is no need to configure the network itself and implement the protocol in a static way. Note that in all cases the Logical Network is going to be different for each and every Mobile Node.
Now let’s 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.

Figure 50: On the left hand side figure one can see the simple, two layered PHMIP protocol, while the right hand side depicts the operation of a multiple layered PHMIP.

4.2.1.3 Personal Tracking Mobile IP – PTMIP

A tracking-like (see Figure 51) 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 [31] or LTRACK [34] protocols. The following message structure would be used for the tracking handover.

```plaintext
[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
 ]
]
```

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.

```plaintext
[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:HA, Action:Delete MN in MAPi-1 via MAPi];
   [Dst:MAPi-2, Src:HA, Action:Delete MN in MAPi-2 via MAPi-1];
   ... //for all former node
 ]
 [Dst: MAPi-1, Src: MAPi,
   Actions: Register MN in MAPi-1 via MAPi]
]
```
There are many proposed methods to decide between the two types of handovers. In my simulation, I implemented a simple suboptimal solution when the MN registers back at every ith step. I analytically compare the results with the one that uses statistical handover decision and with the optimized LTRACK [34].

4.2.1.4 Personal Cellular Mobile IP – PCMIP

Since the widespread use in GSM, 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 search it at each when it is paged. There is a great literature cell forming algorithms. I give an alternative one.

I 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. I 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 tells the IDs of the MAPs in the Paging Range 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 MAP (the leader) the MAPs belonging to that given PR MAPi, MAPj:

\[
\text{[Dst: MAPleader } //\text{The leader of the paging area}//, \text{ Src: MN, Actions: Register MN to MAPleader via MAPi, MAPj, ... , [Dst: HA, Src: MAPleader, Actions: Register MN to HA via MAPleader]}
\]

The problem to solve for cellular algorithms is the problem of forming the 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 what is good enough.
4.2.2 Determine the optimal mobility management strategy

The MN node continuously builds the previously mentioned logical network, consisting of node and edges with different weights, with information gathered from the visited network segment. Based on the logical network stored as an adjacency matrix and the derived network parameters the mobile station always has to be able to determine the most optimal mobility management strategy for itself. MN maintains the following matrix and vector besides the visited logical network:

\[
Q_{\text{Cost}} = \begin{bmatrix}
\infty & C_{\text{strat1} \rightarrow \text{strat2}} & C_{\text{strat1} \rightarrow \text{strat3}} & \cdots & C_{\text{strat1} \rightarrow \text{stratn}} \\
C_{\text{strat2} \rightarrow \text{strat1}} & \infty & C_{\text{strat2} \rightarrow \text{strat3}} & \cdots & C_{\text{strat2} \rightarrow \text{stratn}} \\
C_{\text{strat3} \rightarrow \text{strat1}} & C_{\text{strat3} \rightarrow \text{strat2}} & \infty & \cdots & \cdots \\
\cdots & \cdots & \cdots & \infty & C_{\text{strat}(n-1) \rightarrow \text{stratn}} \\
C_{\text{stratn} \rightarrow \text{strat1}} & C_{\text{stratn} \rightarrow \text{strat2}} & \cdots & C_{\text{stratn} \rightarrow \text{strat(n-1)}} & \infty
\end{bmatrix}, \quad (62)
\]

\[
Q_{\text{Change}} = \begin{bmatrix}
C_{\text{strat1}} & C_{\text{strat2}} & C_{\text{strat3}} & \cdots & C_{\text{stratn}}
\end{bmatrix}
\]

where \(C_{\text{strat1} \rightarrow \text{strat2}}\) is the cost of switching from \(\text{strat1}\) to \(\text{strat2}\), \(C_{\text{tech}}\) is the summarized, possible cost of the tech management strategy. These costs can be calculated pure by the MN (CMFS full control) or the costs can be manipulated by the network providers (CMFS Hybrid mode) in order to get some control over the location management. This matrix and vector can certainly be greater based depending on the number of mobility management strategies known by the mobile.

During its movement if MN reaches a new node or an old one, but updates the \(Q_{\text{LN}}\) matrix with new parameters, the \(Q_{\text{change}}\) matrix and \(Q_{\text{cost}}\) vector is updated accordingly. Then it checks if it is worth to change from the actual management strategy (\(\text{tech}_{\text{actual}}\)) or not:

\[
C_{\text{tech}_{\text{actual}}} < C_{\text{tech}} + C_{\text{tech}_{\text{actual} \rightarrow \text{tech}}}, \forall i. \quad (63)
\]

If the mobile finds a strategy which it should switch to, then it performs the technological handover minimizing and always providing the most optimal mobility support for itself and the network.
4.3 Application of CMFS

CMFS introduced in the previous section is only a framework and a tool to reduce the costs occurring in a network. To be able to reach the mentioned ideal state the appropriate algorithms have to be created in the framework. Using these the mobile devices are able to realize their position management with the least cost for themselves and the network as well.

4.3.1 Tracking-like algorithm

As the first step I have investigated the tracking type position management strategy. The basic question with the tracking solutions is when the MN shall perform a tracking handover and a normal handover (Figure 52). In case of normal handover the mobile entity refreshes it record at the central agent (HA), while in case of tracking handover only at the previous connection point.

The classic LTRACK solution utilizes a Markov chain to determine the optimal number of tracking handovers that are always followed by a normal handover [34]. However this is a global number, all mobiles apply it in a given network. It would be more efficient if all mobiles or a group of mobiles would determine the optimal tracking handover number for themselves. To compare the two theories I have outlined a simple handover situation, where the MN is at MAP_i and moves towards MAP_j (Figure 53). The algorithms I propose require a set of extra data to be recorded by the MN. After switching on, the MN collects data from every network it attached. Let \( N_t = \max K, N \) denote the number of networks visited at time \( t \). The parameters I suggest to record:

- \( u_{i,j} \) – the cost of tracking handover from MAP_i to MAP_j, generally \( u_i \)
- \( d_{i,j} \) – the cost of call delivery from MAP_i to MAP_j, generally \( d_i \)
- \( u_{i} \) – the cost of normal handover from MAP_i to HA
- \( d_{i} \) – the cost of call delivery from HA to MAP_i
- \( p \) – the probability of call delivery to mobile node at MAP_i
- \( r \) – the relative frequency of tracking handover
- \( \lambda_{ij} \): the number of movements from node i to node j
- \( \lambda_j \): the number of movements to node j
- \( \lambda_{ji} \): the number of movements from node j
The \( u_i, u_{ij}, d_i \) and \( d_{ij} \) can be determined from the 'traceroute' times. Let me assume that the distance between MAP\(_{i-1}\) and HA and MAP\(_i\) and HA is equal, therefore \( u_i = u_{i-1} = u \), and \( d_i = d_{i,j} = d \).

These parameters will be used to model the network from the MN point of view, to decide the strategy and the actions. 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 I disregard these factors.

As a result of analysis I have created an alternative, tracking type solution the PTMIP (Personal Tracking Mobile IP).

**Thesis 3.2** I have created the PTMIP (Personal Tracking Mobile IP) tracking-like position management protocol in CMFS. The cost of PTMIP is lower than the cost of LTRACK with \( C_{\text{diff}}^{\text{LT,PT}} \), \( C_{\text{diff}}^{\text{LT,PT}} = \min \{ (1-r)(u_i + d_i, p - u) + up; r(u_i - p_d, p - p_u) \} \) \[ J4,J6,J7 \].

Assume that the MN has just attached to a new MAP and has to decide upon the handovers. I 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 assumptions is very much like the martingale property assumption for stock value changes or call frequency changes in communication networks. To further underpin it I should make measurements on real networks.) Furthermore, if the parameters of the actual MAP are not known then 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 have to be calculated considering possible deliveries too. The expected cost with the tracking handover (at node \( i \)):

\[
c_{\text{track}} = u_{i,prev} + \frac{\mu_i}{\mu_i + \lambda_{\text{corr}}} D_i ,
\]

where \( D_i = d_i + d_{i,2} + ... + d_{\text{prev},i} \) is the total cost of delivery through the chain of tracking points.

The expected cost of a normal handover:
\[ c_{\text{move}} = u_i + \frac{\mu_i}{\mu_i + \lambda_{i-1}} d_i. \quad (65) \]

After the costs recorded the decision is easy to make.

The main idea of PTMIP is that the MN itself decides on performing a tracking handover or a normal handover based on the measured costs. In the classic LTRACK solution, as mentioned previously, the network determines a global tracking handover number valid for all mobiles.

Let us examine one handover generally supposing that it is not a loop (for loops the costs are the same for both protocols). We are at position MAP’ and we move to MAP”’. Then the update and delivery cost from these two access points to the HA are \( u', u'', d', d'' \) respectively. The cost of a tracking handover is supposed to be \( u_t \) and the corresponding delivery is \( d_t \). The experimental probability of receiving a call at MAP”’ is \( p \). We can suppose that the optimal number of tracking handovers for LTRACK is \( r/(1 - r) \) so \( r \) is implicitly defined as the relative frequency of tracking handovers. The expected cost of the LTRACK protocol is then the following:

\[ K_{\text{LT}} := r(u_i + p(d' + d_i + u^*)) + (1 - r)(u^* + pd^*). \quad (66) \]

In the PTMIP strategy the MN calculates the expected cost of the tracing and the normal handover and takes the minimum:

\[ K_{\text{PT}} := \min(K_T := u_i + p(d' + d_i + u^*), K_N := u^* + pd^*). \quad (67) \]

The difference between the two calculated cost \( K_{\text{PT}} \) and \( K_{\text{LT}} \) gives the \( C_{\text{diff}}^{\text{LT,PT}} \) value mentioned in the Thesis. \( C_{\text{diff}}^{\text{LT,PT}} \) only depends on the cost of the update \( u \), it is independent of the call delivery cost, \( d \). The abstract location of the cost functions as a vary of \( u \) is shown in Figure 54.

![Figure 54: The cost functions of \( u \)](image)

Using the situation shown in Figure 54 we can assume that MN moves from MAP_{i-1} to MAP_{i} only knows \( u_{i-1} \) in its current position and visited MAP, first. Using these for the calculation the proper decision is the tracking handover, because at the steady point \( u_{i-1} \) is greater than \( u^* \). The cost of PTMIP will only be greater than the cost of LTRACK, if when recognizing \( u, u < u^* \) and the decision is tracking handover. However we had our base
assumption, that \( u_{i,j} = u_i = u \). Furthermore if MN has previously visited access point MAP, than \( u_i \) is known and calculating with that the decision is obvious.

4.3.2 Cellular-like algorithm

With the widespread use of the modern PLMN technologies the cell-type strategies became more popular. The main point of this solution is that the neighboring cells are organized into a higher level structure, so called paging area (PA), (Location Area in case of GSM, Routing Area in case of UMTS, Tracking Area in case of LTE, Paging Area in case of CIP). The inactive mobiles do not update their location information in case of crossing cell borders, only at paging areas borders. In case of call delivery, as the exact position is unknown, the discovery of the destination station/device is done using broadcast. The idea behind the solution is that the MN moves in a certain set of cells, rarely leaves this area, hence the cost of signaling can be minimized. Grouping and joining the cells like this is an NP-complete problem; more scientific work is dedicated to its solution. However this problem can also be perceived on the level of individuality level, and different PA coverage can be created for each MN.

I gave an algorithm where, based upon a quasi-optimal location area forming algorithm the MN will configure the network to a special CIP or TRACKING-like model [cmfs22]. With this method the MN is able to operate with almost the optimal cost PR-s and signaling 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.

**Thesis 3.3** I have created the PCMIP (Personal Cellular Mobile IP) a cellular-like position management strategy in CMFS, which provides optimal paging area coverage for mobile nodes [J4,J6,J7].

The algorithm works the following way: if the mobile connects to a new access point that is not registered in its own database, it has to decide to add the new access point to the actual PA, create a new PA or reorganize all access points handled by the mobile. The cost of reorganization also has to be taken into account at all steps. The specific cost of maintaining a MAP:

\[
c_i = \mu_d + \sum_{j} \lambda_{j,i} u_i = \mu_d + \lambda u_i .
\] (68)

Now the specific cost of maintaining a new PA, which is the incorporation of MAP into MAP:

\[
c_{i,j} = \mu_j (d_i + d_{i,j}) + (\lambda_j - \lambda_{i,j}) (u_{i,j} + u_i) - \lambda_{j,i} u_i .
\] (69)

Speaking in words, the extra cost when the cell \( j \) is incorporated to cell \( i \) is the incoming call rate \( (\mu_j) \) 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 \( j \) is reached: \( (u_{i,j} + u_i) \), multiplied with this event’s frequency: \( (\lambda_j - \lambda_{i,j}) \), where \( \lambda_j = \sum_{k \neq j} \lambda_{k,j} \). On the other hand, we also save the cost of moving into cell \( i \) from new 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 $M$ subset of all the cells:

$$c_{i,M} = \sum_{n \in M} \mu_n (d_j + d_{i,n}) + \sum_{n \in N \setminus M} \lambda_{n,m} (u_{m,j} + u_i) - \sum_{n \in M} \lambda_{n,m} u_i$$  \hspace{1cm} (70)

where $M$ 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 $c$: MAP $i$);
2. For all $M$ subsets of its (MAP) neighbors, $M \cup N$, including the trivial ones:
   a. Take the subset $J$ of the maximal cost of incorporation.
   b. Take this inequality:
      $$\sum_{j \in J} \lambda_{j,m} + \mu_i d_j < \sum_{j \in J} c_{i,j}$$  \hspace{1cm} (71)

If 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. Before the recalculation algorithm some notations and temporary variables:

- new index determine the new, merged cell (supercell).
- $m \in M$: neighbors merged + initial node, all the nodes in the new, merged cell.
- $n \in N$: neighbors of the new cell, or all the nodes which were not merged and are not in the new cell.
- $n_M$: number of MAPs in subset $M$ (as it was defined earlier).

- $\lambda_{n,m} = \sum_{n \in N} \lambda_{n,m}$
- $\lambda_{n,M} = \sum_{n \in M} \lambda_{n,m}$
- $\mu_M = \sum_{n \in M} \mu_m$
- $p_m = \sum_{n \in N \setminus M} \lambda_{n,m}$ the probability that we arrived to the merged node $m$ from outside of the new cell.
- $\bar{p}_m = \sum_{n \in N \setminus M} \lambda_{n,m}$ the probability that we left the new cell from the merged node $m$.

Then recalculating each cost and parameters:

- $\lambda_{new,m} = \lambda_{i,n} + \sum_{n \in M} \lambda_{n,m}$, \hspace{1cm} (72)
- $\lambda_{n,new} = \lambda_{n,i} + \sum_{n \in M} \lambda_{n,m}$, \hspace{1cm} (73)
- $\mu_{new} = \mu_i + \sum_{n \in M} \mu_m$, \hspace{1cm} (74)
- $u_{new} = u_i + \frac{\sum_{n \in M \setminus N} \lambda_{n,m} u_{n,m}}{n_M}$, \hspace{1cm} (75)
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 normalized with its relative frequency.

$$u_{n,new} = \sum_{m \in M} \frac{\lambda_{n,m}}{\lambda_{n,M}} u_{n,m} .$$  \hspace{1cm} (76)

We sum up the $u_{n,m}$s, but each has to be normalized with the probability of using it:

$$d_{new} = d_i + \sum_{m \in M} \frac{\mu_m d_{i,m}}{\mu_{M}} ,$$  \hspace{1cm} (77)

$$d_{new,n} = \sum_{m \in M} p_m d_{m,n} ,$$  \hspace{1cm} (78)

what is the weighted value of the delivery cost, 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, I still gave a solution good enough.
4.4 Numerical Results

I have made a simulation to show at first that my proposal actually works and secondly to compare it to existing technologies. The simulation was written in the open source OMNet++ [52] 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.

4.4.1 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 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.

![Diagram](image)

Figure 55: The component structure of the simulation of CMFS written in OMNet++.

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 interval 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.

4.4.2 Comparison and analysis of the protocols

In this section I grabbed three main aspects of the new solution to analyze it. At first I show how its costs relates to the simple MIPv4 solution purely collecting network signaling data and examining the whole from QoS point of view. Then I give some numerical results on the performance of a more complex solution, PTMIP. Finally I discuss a few conceptional issues that reveal some fundamental differences between my new protocol and the MIP versus its existing extensions.
I have constructed a virtual test environment consisting of 9 MAs and 9 MNs with the initial MN distribution depicted in Figure 56. This network is very similar to the *Wide-tree topology* (for example an UMTS network, see Figure 11.b) introduced in section 2.3.1. I have chosen this, because the UMTS network is one of the most widely-spread all over the world.

![Figure 56: The test network in OMNet++.](image)

### 4.4.3 Comparing the basic approaches

I have run the simulation on various mobility parameters for all the algorithms separately. All the nodes make 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 in Figure 57. However at low mobility level (when there are only a few handovers between two calls, handovers<10) E-PMIP is better than the classical MIPv4 but as the mobility ratio rises the protocol performs worse in terms of signaling 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 signaling strategy but MIPv4 need Agent Advertisement messages to maintain connectivity while in the client based system can rely on lower layers.

What I 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).
Comparison of the three mobility Management Systems

I have run the simulation again on all types of network from section 2.3.1, with all types of main mobility approach (centralized, hierarchical, wire tracking, wireless tracking and paging type) and with various mobility ratio. I had three different round with 100 MNs. In the first round all mobile chose random mobility approach. In the second round all mobile used the approach which means the lowest cost for the current network type, so the globally optimal solution for the network. In the last round the mobiles worked according to CMFS, thus all mobile used their position management with the least cost for themselves. The signaling traffic of the CMFS solution decreased by 30.13% compared to second, and by 41.08% compared to the first round.

4.4.3.1 Comparison with a QoS-like cost function

I have already presented a figure to compare the approaches from the signaling 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. I construct a very simple kind of QoS cost function.

I can assume that the MN can attach to only one MAP at a time. If not then both protocols could equally benefit from the fact that the MN can receive duplicated messages while changing the MAP. The second assumption is that the signaling 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 then the same method is expected to be applied for the classical and the new solutions with usage of a scale factor $r_d$. As a third assumption, that because of the second one, the transmission speed is in the same linear relation to the logical distance of MAs for both solutions. The parameter $g_T$ (as it was determined earlier) is the distance from the previous MA the MN has attached and the new one. Let us note the average distance from the HA is $m$. Similarly $g_C$ will denote the logical distance to the FA or MAP the MN attaches.

The $t_{MA}^{Protocol}$ and $t_{HA}^{Protocol}$ will denote the average time of processing Protocol messages at the specified nodes.

I measure QoS cost in the relative number of packets to wrong direction for each protocol. I assume that this is a linear function of time thus a linear function of the logical distance and

![Comparison of the three mobility Management Systems](image-url)
processing time. This linear function can be set to the same value 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:

\[
C_{\text{MIP}}^Q = r_d \left( g_C t_{\text{MA}}^{\text{MIP}} + m t_{\text{HA}}^{\text{MIP}} \right),
\]

\[
C_{\text{P-PMIP}}^Q = r_d \left( g_C t_{\text{MA}}^{\text{P-PMIP}} + m t_{\text{HA}}^{\text{P-PMIP}} \right),
\]

\[
C_{\text{E-PMIP}}^Q = r_d \left( g_C t_{\text{MA}}^{\text{E-PMIP}} + g_T t_{\text{MA}}^{\text{E-PMIP}} \right),
\]

where \( r_d \) the scale of the signaling packet delay \((d_d)\) and data packet delay \((d_s)\), and

\[
r_d = \frac{d_d}{d_s}.
\]

Thus we can derive the QoS cost relation between the protocols:

\[
C_{\text{MIP}}^Q - C_{\text{P-PMIP}}^Q = (t_{\text{MA}}^{\text{MIP}} + m t_{\text{HA}}^{\text{MIP}}) - (t_{\text{MA}}^{\text{P-PMIP}} + m t_{\text{HA}}^{\text{P-PMIP}}),
\]

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 \( t_{\text{PROC}} \):

\[
C_{\text{MIP}}^Q - C_{\text{E-PMIP}}^Q = m - g_T.
\]

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.

### 4.4.3.2 Numerical results for complex mobility managements using CMFS

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

Figure 58: This figure depicts the performance 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 Figure 58 one can see that in this case the most tracking handovers performed and thus the most normal handovers avoided significantly decreases the signaling cost for the protocol. Over 30 handover the overhead difference between the approaches (between 'Track 1' and 'Track 3'; between 'Track' 3 and 'Track 5') almost 100%.

4.4.3.3 Differences in the network implementation

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

If I implement Mobile IP to a system, I have to install HA and FA functionality into all the networks I 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 my proposal and there is no need for a complex network structure. The differences appear if I want to extend the algorithms. The classical MIPv4 enhancements such as HMIP [33], TeleMIP [42], DHMIP [31], CMIP [32], Hierarchical Paging [43], LTRACK [34], 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.
5 Conclusion

In this PhD Dissertation I grabbed numerous significant parameters of mobility and proposed a framework to model the mobile node behavior and the network independently of the very technology used. As an example I modeled some general management strategies and obtained technology constants with simulations to give some analytical results. I showed how each modeled protocol responds to tendencies in technology developments and trends and proposed an approach to deal with vertical handover decisions using the framework. With my work it can be identified the mobility management that gives the best solution in different given network scenarios and which aspect of resources could be a bottleneck in each case. This allows operators to tune the parameters of their systems, plan their networks and researchers to propose mobility management solutions of low cost.

In the second main section I enhanced the mobility model. I proposed a classification for Markov mobility models, and I have shown examples for the most important types. I showed the attributes of a general Markov model, and we prepared processes for definition. Obviously these algorithms could be further refined. The Markov mobility model is the most accurate in the estimation process since it has the ability to calculate with motion direction, speed and the recent handover event (user history) also. The three-state model (M3) focuses on cell dwell time since it differentiates only two motion directions which cannot follow general drifts. The n+2-state model (Mn) is sophisticated in both cell dwell time and motion direction since it is capable of following all the different drifts in the cell-cluster.

Using the framework it is not necessary to create a new Markov model, only the description of the network, parameters and the requirement of the accuracy must be given and a Markov movement model is generated with minimal number of states. Using this Markov model the network operator can predict the future distribution and location of users among radio cells to justify CAC or other QoS decisions or support self-configuring system in 4G mobile networks.

At last I have introduced a mobility management system that solves IP mobility from a very different point of view than any other mechanism before. I have shown example algorithms taking ideas from classical solutions. I prepared a simulation and tested my protocol in operation. Using it we compared the performance of some basic solutions and I 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). To discuss the cases when an FA or an MA breaks down or the network could be part of an upcoming work also.

I have shown how CMFS would work over IP. The IMS is gaining importance in today mobile networks, therefore CMFS usage in IMS network would very beneficial, and simple to extend the whole solution to it.
**List of Abbreviations**

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AAEL</td>
<td>Aggregated Average Error Value</td>
</tr>
<tr>
<td>ACM</td>
<td>Access point Centralized Markov model</td>
</tr>
<tr>
<td>ACUM</td>
<td>Access point Centralized Unstructured Markov model</td>
</tr>
<tr>
<td>ACSM</td>
<td>Access point Centralized Structured Markov model</td>
</tr>
<tr>
<td>AP</td>
<td>Access Point</td>
</tr>
<tr>
<td>BA</td>
<td>Binding Acknowledgement</td>
</tr>
<tr>
<td>BR</td>
<td>Binding Request</td>
</tr>
<tr>
<td>BU</td>
<td>Binding Update</td>
</tr>
<tr>
<td>CAC</td>
<td>Call Admission Control</td>
</tr>
<tr>
<td>CIP</td>
<td>Cellular IP</td>
</tr>
<tr>
<td>CMF</td>
<td>Complex Mobility Framework</td>
</tr>
<tr>
<td>CMFS</td>
<td>Client-based Mobility Frame System</td>
</tr>
<tr>
<td>CoA</td>
<td>Care of Address</td>
</tr>
<tr>
<td>DHCP</td>
<td>Dynamic Host Configuration Protocol</td>
</tr>
<tr>
<td>DHMIP</td>
<td>Dinamical Hierarchical Mobile IP</td>
</tr>
<tr>
<td>EPMIP</td>
<td>Extended Personal Mobile IP</td>
</tr>
<tr>
<td>ExtRW</td>
<td>Extended Random Walk</td>
</tr>
<tr>
<td>GRE</td>
<td>Generic Routing Encapsulation</td>
</tr>
<tr>
<td>GSM</td>
<td>Global System for Mobile Communication</td>
</tr>
<tr>
<td>GUI</td>
<td>Graphical User Interface</td>
</tr>
<tr>
<td>HA</td>
<td>Home Agent</td>
</tr>
<tr>
<td>HAWAII</td>
<td>Handoff-Aware Wireless Access Internet Infrastructure</td>
</tr>
<tr>
<td>HIP</td>
<td>Host Identity Protocol</td>
</tr>
<tr>
<td>HMIP</td>
<td>Hierarchical Mobile IP</td>
</tr>
<tr>
<td>HP</td>
<td>Hierarchy Point</td>
</tr>
<tr>
<td>HPAGE</td>
<td>Hierarchical Page</td>
</tr>
<tr>
<td>IMS</td>
<td>IP Multimedia Subsystem</td>
</tr>
<tr>
<td>IP</td>
<td>Internet Protocol</td>
</tr>
<tr>
<td>IRDP</td>
<td>ICMP Router Discovery Protocol</td>
</tr>
<tr>
<td>LA</td>
<td>Location Area</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>------------------------------------</td>
</tr>
<tr>
<td>LAN</td>
<td>Local Area Networks</td>
</tr>
<tr>
<td>LN</td>
<td>Logical Network</td>
</tr>
<tr>
<td>LTE</td>
<td>Long Term Evolution</td>
</tr>
<tr>
<td>LTRACK</td>
<td>Location Tracking</td>
</tr>
<tr>
<td>MA</td>
<td>Mobility Agent</td>
</tr>
<tr>
<td>MANET</td>
<td>Mobile Ad Hoc Network</td>
</tr>
<tr>
<td>MAP</td>
<td>Mobility Access Point</td>
</tr>
<tr>
<td>MIP</td>
<td>Mobile IPv4</td>
</tr>
<tr>
<td>MIPv6</td>
<td>Mobile IPv6</td>
</tr>
<tr>
<td>MMCS</td>
<td>Markov Movement-model Creator System</td>
</tr>
<tr>
<td>MN</td>
<td>Mobile Node</td>
</tr>
<tr>
<td>mSCTP</td>
<td>multi-homing Stream Control Transmission Protocol</td>
</tr>
<tr>
<td>NE</td>
<td>Network Element</td>
</tr>
<tr>
<td>NGN</td>
<td>Next Generation Network</td>
</tr>
<tr>
<td>OoS</td>
<td>Quality of Service</td>
</tr>
<tr>
<td>PCMIP</td>
<td>Personal Cellular Mobile IP</td>
</tr>
<tr>
<td>PHMIP</td>
<td>Personal Hierarchical Mobile IP</td>
</tr>
<tr>
<td>PMIP</td>
<td>Personal Mobile IP</td>
</tr>
<tr>
<td>PPMIP</td>
<td>Pure Personal Mobile IP</td>
</tr>
<tr>
<td>PTMIP</td>
<td>Personal Tracking Mobile IP</td>
</tr>
<tr>
<td>RA</td>
<td>Routing Area</td>
</tr>
<tr>
<td>RPGM</td>
<td>Reference Point Group Mobility</td>
</tr>
<tr>
<td>RW</td>
<td>Random Walk</td>
</tr>
<tr>
<td>SDT</td>
<td>Standard Deviation of the TVEL</td>
</tr>
<tr>
<td>SIP</td>
<td>Session Initiation Protocol</td>
</tr>
<tr>
<td>TA</td>
<td>Tracking Area</td>
</tr>
<tr>
<td>TAEV</td>
<td>Time-dependent Average Error Value</td>
</tr>
<tr>
<td>TCP</td>
<td>Transmission Control Protocol</td>
</tr>
<tr>
<td>TDOH</td>
<td>Tunneled Destination Option Header</td>
</tr>
<tr>
<td>UCM</td>
<td>User Centralized Markov model</td>
</tr>
<tr>
<td>Acronym</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-----------------------------------------</td>
</tr>
<tr>
<td>UDP</td>
<td>User Datagram Protocol</td>
</tr>
<tr>
<td>UMTS</td>
<td>Universal Mobile Telecommunications System</td>
</tr>
<tr>
<td>URA</td>
<td>UTRAN Registration Area</td>
</tr>
<tr>
<td>UTRAN</td>
<td>UMTS Terrestrial Radio Access Network</td>
</tr>
<tr>
<td>WLAN</td>
<td>Wireless Local Area Networks</td>
</tr>
</tbody>
</table>
Publications


[J7]. P. Fülöp, B. Kovács, “Numerical analysis of mobility management algorithms”, Info-Communications-Technology, Volume LxII., pp. 32-38, 2007/7,


References


[23]. Frédéric Lassabe, Damien Charlet, Philippe Canalda, Pascal Chatonnay, and François Spies, “Predictive Mobility Models based on Kth Markov Models” In IEEE Int. Conf. on Pervasive Services 2006 (ICPS'06), Lyon, France, pp 303-306, June 2006.


[52]. OMNeT++ Discrete Event Simulator http://www.omnetpp.org/


[81]. Floyd algorithm: https://en.wikipedia.org/wiki/Floyd%E2%80%93Warshall_algorithm


