Movement modeling in cellular networks - Markov Movement Model Creator Framework (MMCF)

Péter Fülöp, Dr. Sándor Szabó

***Abstract-* Nowadays, in the wireless networks the number of users and the transferred packet switched data speed are increasing dramatically. Due to the demands and the market competition the services are becoming more and more complex.**

The efficient network dimensioning and configuration highly depends on the underlying mathematical model of user distribution and expected data transfer level. In this paper we propose a Markov Movement-Model Creator Framework (MMCF) for setting up a model based on the network parameters and requirements with optimal number of states.

Firstly we describe a method that yields an abstract model of the mobile network and the node, and we introduce a simple classifying method that defines the necessary parameters of the exact Markov movement model. The mathematical solutions for determining these parameters are also presented in the paper. Finally we analyze the accuracy, complexity and usability of the proposed MMCF and an analytical comparison is made with other mobility models, the comparison is proved by simulations. The movement model created with the framework helps the network operators in setting up an effective authorization, fraud detection system or solving self-configuration issues.

***Index Terms*-Movement model; Framework; Markov; Accuracy; Complexity; User movements;**

I. INTRODUCTION

The number of users and the amount of transferred data is increasing dynamically and substantially in the mobile networks. There are also more and more new technologies, standards (for example HSXPA-High Speed Packet Data, LTE-Long Term Evolution [1]), and future solutions [2][11] to support efficient mobility. Hence the network providers and operators face more and more complex management systems and operation tasks. Wireless multimedia and other services have many requirements and the resources in the network are often expensive and limited. Nowadays the operation tasks have some critical parts, i.e. guaranteeing the security of user-related information and data, providing QoS (Quality of Service), location management and maintaining the service levels in the network.[1]

In recent years people increasingly rely on wireless devices in their daily life for very sensitive tasks such as shopping and bank transactions. Although many authentication protocols are used in wireless, mobile networks, it is still a challenging task to design a fully secure mobile environment because of the open radio transmission environment and the vulnerability of mobile devices. Anomaly-based detection as part of the detection-based techniques creates normal profiles of system states or user behaviours, stores and periodically compares them with the current activities. If significant deviation is detected, the network system raises an alarm. A user profile is very difficult to build up, but it could largely increase the security of a wireless system [4][5]. Movement mobility models and location prediction take significant parts in creating user profiles as well.

The previously mentioned fast evolution of network applications and services require skilled individuals to install, configure and maintain these systems. Other possibility is to introduce mechanisms and procedures, which enable a system to reconfigure, heal or install itself [6]. These systems shall be capable of modifying their own behaviour and adapting environmental changes based on performance measures. A well developed movement mobility model can be used as a proper trigger to rebuild a cell boundary in 4G LTE (Long Term Evolution) network, recognize radio interface problem of mobile access points [1].

Different areas have been introduced above, where mobility models should be used, but the scope of this research area is far more wider nowadays. The discussion of individual or group mobility modelling have been addressed in many papers in the literature [5][7][8][9][10][12][23]. Beside other approaches a few propositions are using Markov model as a sophisticated mathematical solution [8][9][10][13]

As highlighted above an efficient mobility movement model is necessary for network providers nowadays. In this paper our aim is to give general design guidelines to create Markov movement mobility models with optimal number of states and proper accuracy according to the network and user movement parameters.

The structure of the paper is the following. In Section II we give a brief on the mobility models. The model for the network and the mobile node with its mobility parameters are introduced in Section III. This is followed with the definitions of classifying system for Markov movement models. In Section V the Markov Movement model Creator Frameworks is introduced and analysed, while Markov model examples are derived in Section VI. Finally simulation and numerical results are given in Section VII.

II. RELATED WORK

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 section we give a short overview of mobility models.

In the traditional Random Walk Mobility Model the node moves from its current location to a new contiguous location by randomly choosing a direction and a speed. The Random Walk Model defines user movement from one position to the next with randomly selected speed and direction. Many derivatives of the Random Walk mobility model have been

---

Manuscript submitted April 2013, revised May 22, 2013.

Péter Fülöp, Department of Telecommunication, Budapest University of Technology and Economics Magyar tudósok körútja, Budapest 1117, Hungary, fpeti@mcl.hu

Dr. Sándor Szabó, Department of Telecommunication, Budapest University of Technology and Economics Magyar tudósok körútja, Budapest 1117, Hungary, szabos@hit.bme.hu

developed including one, two, three - and d-dimensional walks [20].

A flexible mobility framework for hybrid motion patterns is the Mobility Vector model [7]. A mobility vector expresses the mobility of a node as the sum of two sub-vectors: the Base Vector (BV) $\vec{B} = (bx_v, by_v)$ and the Deviation Vector (DV) $\vec{V} = (vx_v, vy_v)$. The BV defines the major direction and velocity of the node while the DV stores the mobility deviation from the base vector. The mobility vector $\vec{M}$ is expressed as $\vec{M} = B + \alpha V$ where α is an acceleration factor.

The location history of a mobile user is exploited in High-Order Markov Model that is described in [5][8]. 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. [9] 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. They present two predictive models from the AKMM: the K-to-1 past Model and its improvement, the K-to-1 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 [14]. 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 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.

A ring-based mobility prediction and resource reservation algorithm is proposed in [10]. 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 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 [14] to an inter-cell level.

W. Ma et al propose a user mobility pattern (UMP) based model (Mobility Pattern-Based Scheme – MPBS) in [16]. 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 which RA is the user located in at the actual timestamp. Naturally, the ideal user who always follows the timeline does not exist, but the time-sequence and RA-sequence provide information even if the actual timeline differs from the pre-recorded one. A categorization is presented with four categories where the first category is the ideal user with a timeline-compatible trajectory. The 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 pre-recorded ones.

III. Modelling the Network

In this section we collect the most significant properties and parameters of the mobile network that can describe an abstract network model.

A. Basic notations and descriptions

We define the basic notations that we use is the article. The basic model will resemble to the abstract one in our previous work [16][17].

- The specific network with all its parameters is denoted with N.
- The Mobile Nodes (MN, alias mobiles, moving entities, users) are the mobile equipments that want to communicate with other mobile nodes or fixed partners and move between the radio access points. The number of users (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 of communicating with the mobile nodes via radio interface. All mobility access points have their own geographical areas. While the MN moves in an area, it is always connected to the owner of the 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 neighbouring MAPs, each user is connected to only one MAP at a time. The neighbour MAPs could use even different access technologies than the current MAP and they could be located in the very same geographical place as well, the model does not require single access technology in the whole network. The number of neighbour MAPs of MAP_i is denoted by n^i_{nm}.
- There are other network elements which provide the communication in the core network behind the MAPs. We denote these as Network Elements (NE).

- Network trace is an abstraction of the network operation log, it contains 4-tuples of a timestamp, user ID, MAP ID and network event. A trace entry could mean for example that the selected user connected with handover to the MAP at the given timestamp. The network trace contains all information of the mobility of the users in the network.

B. *Deriving parameters of a given network*

A way to describe a network is to observe the network trace. We 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 state change of the given user (e.g. mobile node is in idle status, voice call or data transfer is set up, cell boundary crossing). The logical location of the event is determined by the MAP ID where the user is located at the timestamp of the event. (Table 1). 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 1. AN EXAMPLE OF WIRELESS NETWORK MANAGEMENT SYSTEM'S LOG

Timestamp	User ID	MAP id	State or Event
...			
09:21:43:12	41	4951	Idle
09:21:43:12	41	4957	Idle
09:21:43:12	41	4957	Voice call
09:21:43:12	19	5341	Data Call/Traffic class2
09:21:43:12	84	7120	Idle
09:21:43:12	19	5348	Data Call/Traffic class2
09:21:43:12	19	5348	Idle
...			

The aim is to derive the parameters of the user mobility, therefore we should pick the relevant entries from the network trace. In our work we 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.

We assumed that the user distribution in the network is given at the first moment of the reference interval. We created a discrete sample series where samples are taken at Δt time intervals, that is a location state is assigned to every users per Δt time. Δt is defined system-wide as the minimum of the time intervals 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 state reports are processed. With this sampling a MAP ID and a state can be determined to every user in every timeslot. The sampling results a n_u x 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 state of each user in each timeslot.

The relative frequency of any state 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. We defined the S set, which contains all possible states and events appearing in the logs. The important ones are the following:

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

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

To determine the relative frequencies of states in a MAP, the state must occur frequently enough, otherwise it is neglected. Let us define the n_u x n_m $C_s=[c^s_{i,j}]$ matrix, where $c^s_{i,j}$, $s \in S$, is the occurrence of s state with user i, in MAP j from the P matrix. The average occurrence of s state in the network is:

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

The parameter c_s can be used as a main requirement in order to create a valid model based on the network description. So the network must be monitored for sufficient time before we create a model from it.

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

$$D^s = [d^s_{i,j}] = \begin{cases} 0, & c^s_{i,j} < \varepsilon_C \\ \dfrac{c^s_{i,j}}{c^s_j} \end{cases} \qquad c^s_j = \sum_{\forall i} c^s_{i,j} \;\; s \in S \quad (2)$$

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

We determine the rate of receiving a call with μ. It can be determined for every MN in every MAP from the $D^u=[\mu_{i,j}]$, but the average value is also calculable in similar manner described above:

$$\mu = \frac{\sum_{\forall i}\sum_{\forall j} \mu_{i,j}}{n_u \cdot n_m} \quad (3)$$

Let us have the corresponding network graph given with its weighted adjacency matrix: A.

Let us assume that the aggregated behaviour of the Mobile Nodes can be modelled 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. [18]). The chain is given by a rate matrix $B_Q = [b_{ij}]$. In this matrix, all the possible MAP-s are listed so the matrix will be a n_m x n_m matrix where each element b_{ij} denotes how frequent the movement of the mobile is from $\mathrm{MAP}_i \rightarrow \mathrm{MAP}_j$. 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_Π can be determined easily. We assume that the matrix B_Π, 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 $\underline{b}$. 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 $B^{i,s}_\Pi$ 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:

$$n_{vm} = \underline{1} \cdot sign(B_\Pi \cdot \underline{1}) \qquad (4)$$

A general network describing parameter, the weighted average of visited MAPs is:

$$w_{vm} = \frac{\sum_{\forall i} \underline{b} \cdot sign(B^i{}_\Pi \cdot \underline{1})}{n_u} \qquad (5)$$

The other parameter is the average number of neighbouring MAs that can be accessed via a wire from a given node: w_{nm}. It should be also weighted with the probability density of the MN.

$$w_{nm} = \frac{(\underline{b} * sign(B_\Pi)) \cdot \underline{1}}{n_a} \qquad (6)$$

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

IV. Classifying Markov movement models

In this section, a simple classification of motion models is presented. The aim is to compare different models easily and to analyse them in different network environments. The purpose 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.

We used a discrete-time, finite-state and infinite-state Markov-chain to model the mobile movement.

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)* as a 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. 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 models are distinguished, the *User-Centralized* and the *Access Point Centralized* model. The latter one is further separated into two subtypes. Figure 1 depicts this main classification.

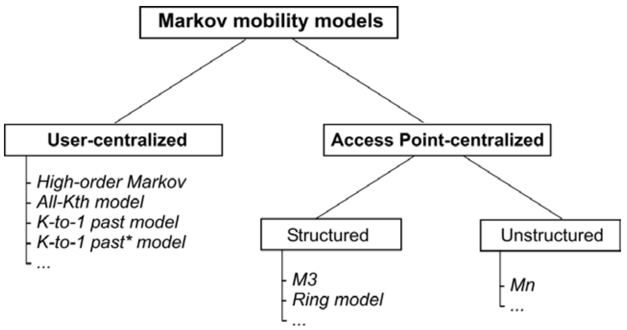


Fig. 1 The general classifying of the Markov mobility models

In the next subsection these groups of models are explained.

A. 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 behaviour is modelled 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.

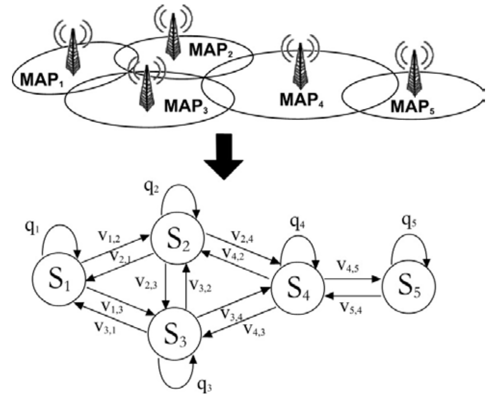


Fig. 2 The creation of user-centralized models

Figure 2 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_Π transition matrix.

This usage of this model is reasonable, if the behaviour 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 [5], [9].

B. 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 modelling the behaviour of an individual user, a MAP and its environment is to be observed. In these cases a MAP or more MAPs and their defined neighbours 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, this creating 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-sections.

Structured Markov Models (ACSM)

In the structured Markov model groups of MAPs are defined. The grouping can be derived from user behaviour, geographical specialty or even from network requirements. Figure 3 shows examples for structured solutions, the Ring Model (*RM*) [10] and the *M3* model [13].

In the *RM* (Figure 3.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 [10].

The *M3* model handles users in four different Markov states, the right-area, left-area, stay and outside state. More details about the M3 model are given in Section VI.A [13].

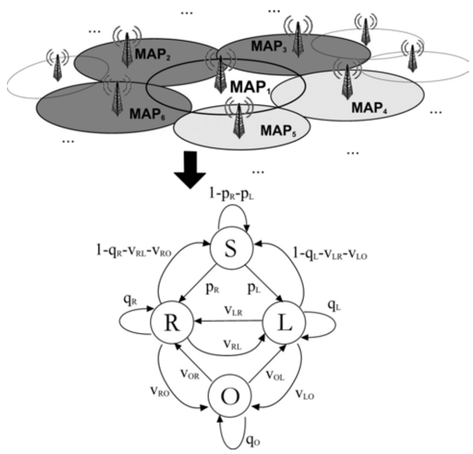


Fig. 3.a The representation of access point-centralized/structured models, M3 model [13]

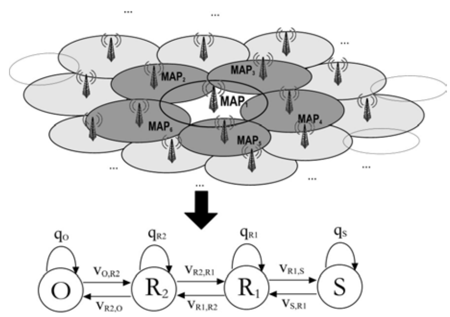


Fig. 3.b The representation of access point-centralized/structured models, Ring model [10]

Let us introduce the theoretical error E_T, which is the sum of error percentages per MAP in prediction. 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.

Unstructured Markov Models (ACSM)

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 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.

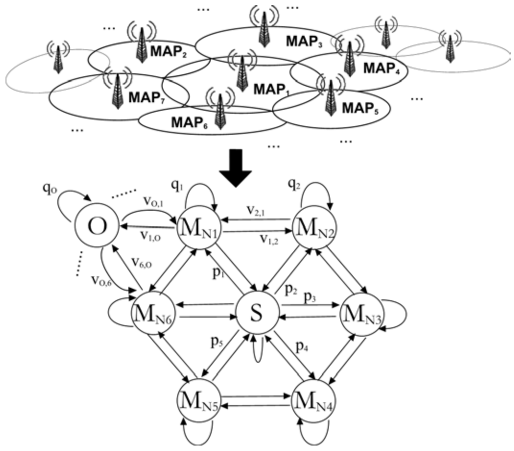


Fig. 4 Access point-centralized/Unstructured/M7 model [13]

Figure 4 depicts the methodology of unstructured model representation. The Markov chain in Figure 4 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.

C. *Attributes*

The example models introduced above are the simplest ones in their class. In this section we determine attributes to the classifying system which describe important parameters 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.

Level of the model

As mentioned in Section III.B, B_Π could be determined from the *P* matrix for every state as well. There are two main reasons to handle the states differently:

-the users behave differently in certain states,

-the users in distinct states must be modelled in a different way (for example different CAC is used for the users in voice call, than the users downloading data from the internet).

In these cases the B_Π must be calculated for different states. This diversity in the model is represented by 'levels' or 'dimension' (Figure 5). The transition rates between the levels show the intensity state changes in the current MAPs. This model is similar to the one described in [19].

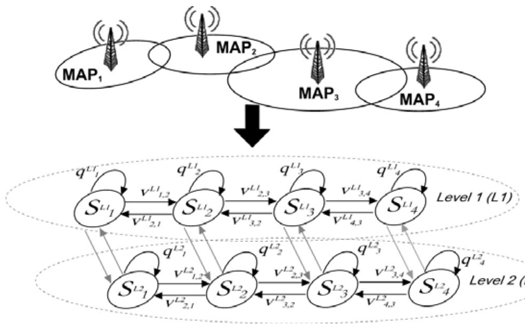


Fig. 5 Attributes in Markov mobility model classifying: Example for meaning of 'level'

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 $\underline{L} = [L_1,...,L_{n_L}]$. A specific L is based on its $B^s{}_{\Pi}$ matrix.

Resolution of the model

There is a possibility to merge adjacent MAPs together, if those MAPs are not needed to be handled separately. If outgoing predictions of users in two adjacent MAPs match within a certain limit, the two adjacent MAPs could be merged together and handled henceforward as a new major MAP. By this the complexity of the model can be decreased. Figure 6 shows an example, in which the 6 neighbour MAPs of an access point-centralized, unstructured model, are merged into 3 new major MAPs.

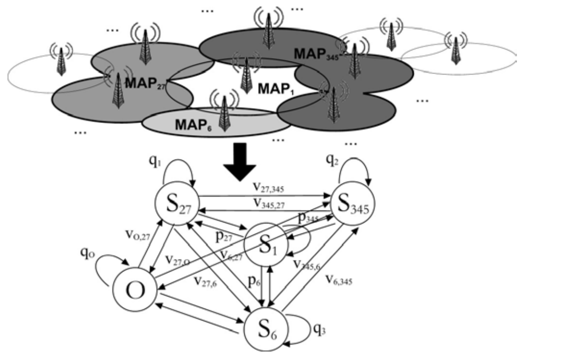


Fig. 6 Attributes in Markov mobility model classifying: Example for meaning of 'resolution'

'Grouping' explained in Section IV.B.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.

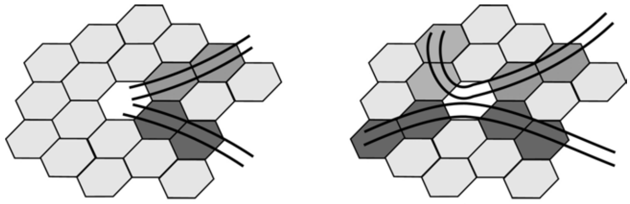
Every level could have its own resolution. The resolution is denoted with R, $R = \{G_1,...,G_{n_M}\}$ where G is a set of merged MAPs, and n_M the number of new MAPs after merging. The R is described with a general rate, n_m:n_M. The vector $\underline{R} = [R_1,...,R_{n_L}]$ contains the resolution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

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) [13].

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.

We present a simple example which shows the effect of depth or memory in the model in our previous work [13]. If we consider the two roads shown is Figure 7.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 7.a.).



a. b.

Fig. 7 User prediction methods
a. model without using *memory*, unknown where is the users come from
b. model with *memory*, the previous steps of the users taken in account

The results show that our 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 our Markov models similar like in [5]. In our model, sequence of MAP IDs can be assigned to every MAP not to a user; ID_1, ID_2, . . ., ID_i, . . . , 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 into 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 a more specific MAP ID list could belong to a MAP, and for this reason more Markov state assigned to a MAP, see Figure 9.

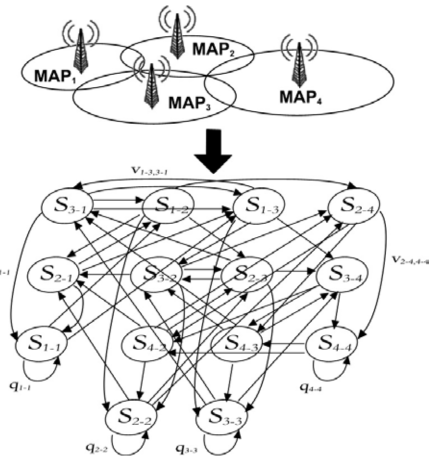


Fig. 9 Example for meaning 'memory', 2nd model [9]

Thus $\underline{O} = [o_1,...,o_{n_m}]$ matrix denotes the depth of the model for a level, where o_i is the applied sequence length of previous visited MAPs to MAP$_i$. Generally the n_m x n_L O matrix ($\underline{\underline{O}} = [\underline{O}_1,...,\underline{O}_{n_L}]$) belongs to a Markov model.

The weighted average depth for a model is the following:

$$w_O = \frac{\sum_{i=1}^{n_L} \underline{b}^i \cdot \underline{O}^i}{n_L} \qquad (7)$$

Complexity

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

$$n_{states} = \sum_{\forall l \in L} n^{l}{}_{m} / n^{l}{}_{M} \cdot {w^{l}{}_{vnm}}^{w^{l}{}_{O}} \qquad (8)$$

D. *Performance analysis*

In this section we present a simple performance analysis between the Markov models introduced in the previous sections. The models are examined in different network environments. In every scenario the analysis is performed on a cluster of 7 hexagonal radio cells (Figure 10). In our interpretation the performance of the mentioned Markov models depends on the theoretical error (E_T) and the number of states (n_{states}), in this case a special theoretical cost $C_{MM}=f(n_{states},E_T)$. The lower this cost is, the better the performance. The computational capacity increases, hence the steady state probabilities determination of Markov models with more states are less difficult. But the theoretical error in prediction is more important. For these reasons we calculate the theoretical cost this way:

$$C_{MM} = \log n_{states} (1+E_T)^2 \qquad (9)$$

where n_{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, which is the sum of error percentages per MAP in prediction.

The following models were examined in the performance analysis:

-RM: The model introduced in Section IV.B.1 (Figure 3.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_1-MAP_6.

-M3: Also introduced in Section IV.B.1 (Figure 3.a.). MAP_2, MAP_3 and MAP_7 belong to the left-area state (L). The right-area state (R) includes MAP_4, MAP_5, and MAP_6.

-M7: The model presented in Section IV.B.2 (Figure 4).

-2nd: The extension of model M7, the memory is increased to 2. It is similar to 2nd model in Figure 9.

-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.

We investigated special network environments (scenarios) to highlight the advantages and disadvantages of each model. The different network scenarios are shown in Figure 10. These are the following cases:

- Scenario a.: 'A park, uniform user distribution'
- Scenario b.: 'Simple road'
- Scenario c.: 'Highway to city'
- Scenario d.: 'Directional motion'
- Scenario e.: 'Differentiated users'

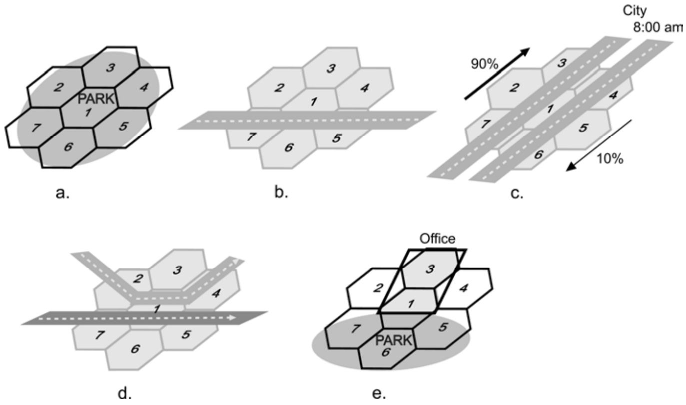


Fig. 10 The network scenarios for performance analysis

TABLE 2. THE RESULT OF PERFORMANCE ANALYSIS

C_{MM}	a.	b.	c.	d.	e.
RM (n_{states}=3)	*0,47*	*2,55*	*1,52*	*2,55*	*2,55*
M3 (n_{states}=4)	*0,6*	***3,2***	*0,6*	*3,2*	*3,2*
M7 (n_{states}=8)	*0,9*	*0,9*	*0,9*	*3,6*	*3,6*
2nd (n_{states}=49)	***1,69***	*1,69*	***1,69***	*1,69*	***6,76***
3D (n_{states}=16)	*1,2*	*1,2*	*1,2*	***4,8***	*1,2*

Table 2 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 E_T 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 (MAP_7, MAP $_2$, MAP $_3$), while all of them stay in *M7*. Therefore the model has 133,3% 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 in 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 neighbor MAPs.

Scenario d. is very similar to the example presented in Section IV.B.3.3. 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 have 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 that each model performs best in specific situations, accordingly it is important to choose the proper model in every network scenario.

V. Markov movement-Model Creator Framework - MMCF

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

If every MAP is handled separately and all previous steps, user groups are taken into account then the modelling Markov chain will contain a high number of states. Therefore a minimal error rate should be allowed and the appropriate Markov motion model with the minimal number of states should be found.

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

We proposed a general Markov Movement model Creator System (MMCS). For every $N<A,D,B>$ network, where A is the adjacent matrix of the network, D is the occurrence matrix and B_Π is the handover matrix and for ε error vector, an optimal number of states $M<L,R,O>$ Markov model can be determined, where L is the level-, R is the resolution-, and O is the depth of the model.

A. Error vector

The error vector is an input for the MMCF. The elements of the vector are as follows:

- ε_{ds} the limit of mean difference between the transition matrixes for different user states.
- ε_{rd} the acceptable error rate of outgoing prediction probability, when MAPs are merging together.
- ε_{op} the limit of the difference of the outgoing probabilities of the sequences from current MAP.
- ε_{uv} the limit of fluctuation of the number of users arriving from a certain sequence directions is investigated .

More details about the error rates are presented in V.C, V.D and V.E Sections.

B. 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
  - Modelling from user point of view
  - User profile creation
  - Fraud detection
- Access Point-Centralized
  - Modelling from cell point of view
  - CAC in a MAP
  - Movement modelling of a geographical area

C. Determining the level

A new level should be applied in the model, if the mean difference between transition matrices for different user states 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 ε_{ds} as the limit of the mean difference between the B_Π and transition matrix for different s user states ($B^s{}_\Pi$). The average weighted deviation can be calculated by the following form:

$$w^s{}_{ds} = \frac{\underline{b} \cdot abs(B_\Pi - B^s{}_\Pi) \cdot \underline{1}}{n_u^{\;2}} \quad s \in S \qquad (10)$$

If $w^s{}_{ds} > \varepsilon_{ds}$ is true for a state s, then a new level must be introduced into the model for state s. This inequality must be analysed for all $s \in S$.

The number of levels can be calculated as follows:

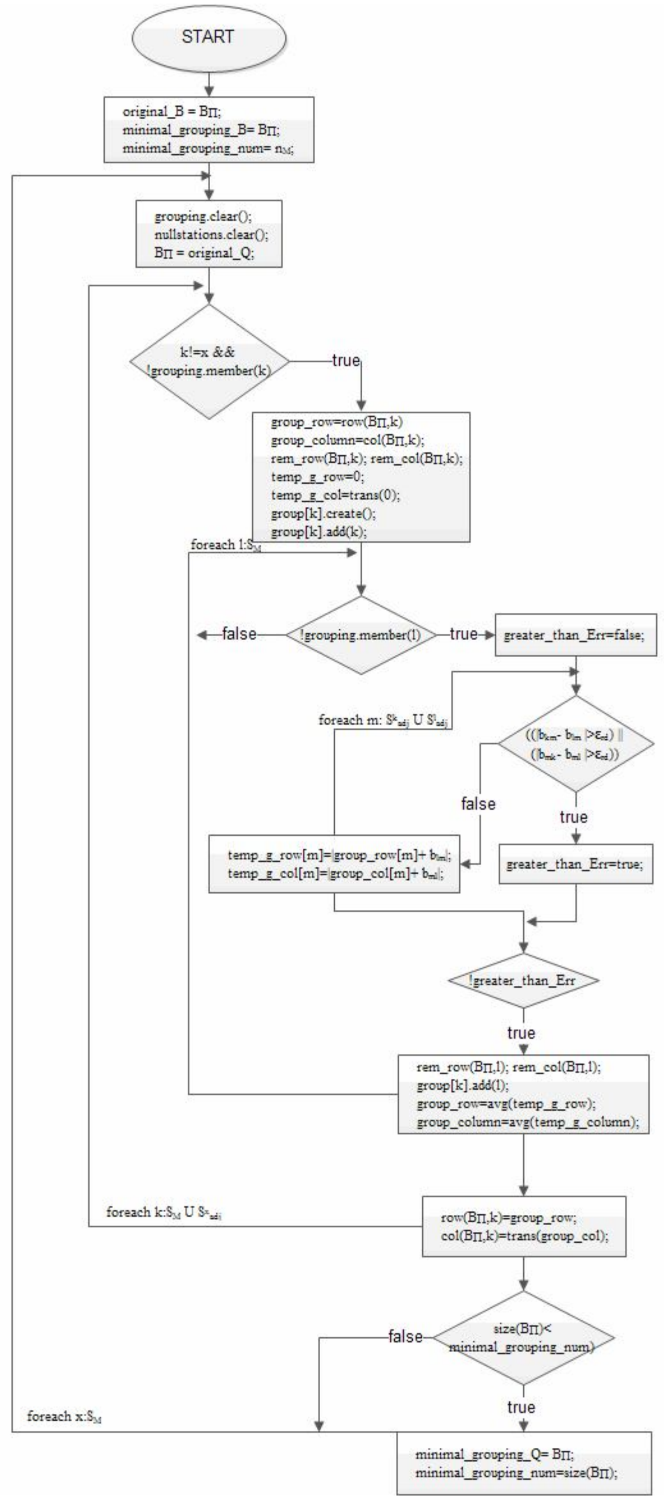
$$n_L = \sum_{\forall s \in S} sign(v^s{}_{ds} + abs(v^s{}_{ds})), \qquad v^s{}_{ds} = w^s{}_{ds} - \varepsilon_{ds} \quad (11)$$

D. Determining the resolution

As we explained earlier if outgoing predictions of users in more adjacent MAPs match 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.

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

- ε_{rd}, the acceptable error rate of outgoing prediction probability, when MAPs are merging together,
- transition matrix, B_Π , as it was defined earlier,



where S_M the set of MAPs in the model, S^a_{adj} 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 in a smaller transition matrix, B_Π*, which leads to a simpler model.

E. 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 9, where the depth is generally 2, for all MAPs ($\underline{Q} = [2,2,2,2]$).

The main idea is to analyse 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 ε_{op} as a limit for this difference. The difference for a MAP_i and k-depth is determined the following way:

$$Df[op]_i^k = \frac{\sum_{b \in Q_i^k} \sum_{a \in Q_i^k} \frac{\sum_{l \forall NB_i} \left| b_{q(a),l} - b_{q(b),l} \right|}{n^i{}_{nm}}}{(w_{vnm}{}^k)^2} \tag{12}$$

where Q_{ik} is the set of existing *k* length sequences from MAP_i, NB_i is the set of neighbour MAP IDs of MAP_i and *q* denotes a sequence from the set.

The first criterion of importance is: $Df[op]_i^k > \varepsilon_{op}$.

- 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 ε_{uv} as a limit for this variance. The variance of number of incoming users from a sequence into MAP_i:

$$V_i^a = E\left(\left| E\left(\frac{n^{i,a}{}_u}{n^i{}_u} \right) - \left(\frac{n^{i,a}{}_u}{n^i{}_u} \right) \right| \right) = \sigma \left(\frac{n^{i,a}{}_u}{n^i{}_u} \right) \tag{13}$$

where $n^{i,a}{}_u$ is the number of users in MAP_i arrived from a sequence (path), $n^i{}_u$ is the number of users in MAP_i.

This must be examined for all of the incoming sequences:

$$Df\,[uv]_i^k = \frac{\sum_{a \in Q_i^k} V_i^{q(a)}}{w_{vnm}{}^k} \tag{14}$$

The second criterion of importance is: $Df[uv]_i^k > \varepsilon_{uv}$.

This two criteria, $Df[op]_i^m > \varepsilon op$ and $Df[uv]_i^m > \varepsilon_{uv}$ must be applied for all MAP in order to determine $\underline{Q}$.

VI. Markov Model Examples

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

A. The M3 model

In the *M3*, Markov-chain based model: 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) (Figure 3.a).

The grouping can be derived from the user behaviour. If the users in right-hand side cells behave similarly from the current cell's point of view, the neighbouring 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 [13].

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

The *M3* model could be determined as an MMCF model, with the following parameters:

n_L=1, $\underline{\underline{O}}$=[1,1,1], $\underline{\underline{R}}$ =[1,{2,3,4},{5,6,7}].

B. Generalized Mn model

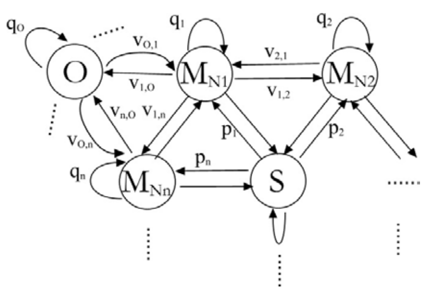
We have enhanced the access point-centralized Markov model, *M3* to generalized *n+2* state Markov model (*Mn*).

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 neighbour cells as a separated Markov state.

As we described above, in the unstructured models a MAP is simply mapped into a Markov-chain state, so 8 states are created, because 7 elements are assumed in a theoretical cluster (direction dimension is 6) and another one to the outside world. This results the M7 model [13].

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. We expanded our previously mentioned model to n-neighbour case (Figure 11), when all the n neighbours are represented with a Markov state:

- stationary state (*S*)
- neighbour 1...n state (M_{N1}... M_{Nn})
- outside area state (*O*)



$$\Pi = \begin{bmatrix} 1-\sum_{m=1}^{n} p_m & p_1 & p_2 & \dots & \dots & \dots & p_n & 0 \\ 1-q_1-v_{1,2}-v_{1,n}-v_{1,O} & q_1 & v_{1,2} & 0 & \dots & \dots & v_{1,n} & v_{1,O} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 1-q_i-v_{i,i-1}-v_{i,i+1}-v_{i,O} & 0 & \dots & v_{i,i-1} & q_i & v_{i,i+1} & \dots & v_{i,O} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 1-q_n-v_{n,1}-v_{n,n-1}-v_{n,O} & v_{n,1} & \dots & \dots & 0 & v_{n,n-1} & q_n & v_{n,O} \\ 0 & v_{O,1} & v_{O,2} & \dots & \dots & v_{O,n-1} & v_{O,n} & q_O \end{bmatrix}$$

Fig. 11 State diagram and Π matrix of n+2-state Markov model (*Mn*)

The steady state probabilities (*Ps*, P_{N1}...P_{Nn}, P_{NO}) can be calculated.

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 Eq. 15.

$$\begin{aligned} N_i^{t+1} &= N_i^{\,t} \cdot (1-\sum_{m=1}^{n} p_m(i)) + \\ &+ \sum_{j,MAP_j \in S^i_{adj}} N_j^{\,t} \cdot (1-q_j(i) - v_{j,j+1 \bmod n}(i) - v_{j,j-1 \bmod n}(i) - v_{j,O}(i)) \qquad (a) \\ &\dots \\ N_k^{t+1} &= N_k^{\,t} \cdot q_k(i) + N_i^{\,t} \cdot p_k(i) + \sum_{j,MAP_j \in (S^k_{adj} \cap S^i_{adj})} N_j \cdot v_{j,k}(i) + \\ &+ N_O^{\,t} \cdot v_{O,k}(i) \qquad k \in S^i_{adj} \qquad (b) \end{aligned} \tag{15}$$

Because all of neighbour MAPs is represented as separated states, the prediction based *Mn* model is more accurate than the *M3* model, or at least as accurate as model *M3*.

The inaccuracy of prediction N_i^{t+1} from MAP_j is the following:

$$\left(\frac{1}{3} - \frac{N_j^t}{N_K^t}\right) N_K^t \cdot b_{j,i} \tag{16}$$

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:

$$\sum_{K=L,R} \frac{\sum_{j,MAP_j \in S_k} \left\{ \left| \frac{1}{3} - \left(\frac{N_j^t}{N_K^t} \right) \right| N_K^t \cdot b_{i,j} \right\}}{N_K^t} \tag{17}$$

Let us determine the inaccuracy in a different way. The difference between the two predictions, *M3* and *Mn*, weighted with the real number of users:

$$\frac{N^{t+1}_{i\ M3} - N^{t+1}_{i\ Mn}}{N_i^{t+1}} \tag{18}$$

Instead of transition probabilities *p*, *q*, *v* the elements of the rate matrix $B_Q = [b_{ij}]$ are used. The Eq. 19 shows the proof, which follows the same result as in Eq. 17:

$$\begin{aligned} &\left(N_i^{\,t} \cdot b_{i,i} + N_L^{\,t} \cdot b_{L,i} + N_R^{\,t} \cdot b_{R,i}\right) - \left(N_i^{\,t} \cdot b_{i,i} + \sum_{j,MAP_j \in S^i_{adj}} N_j^{\,t} \cdot b_{j,i} \right) \\ &\left(\sum_{K=L,R} \sum_{j,S^i_{adj(K)}} N_j \cdot b_{j,i} \right) - \left(\sum_{j,S^i_{adj}} N_j^{\,t} \cdot b_{j,i} \right) = \\ &\sum_{K=L,R} \left(\sum_{j,S^i_{adj}} \left(\frac{1}{3} - \frac{N_j}{N_k} \right) N_k \cdot b_{j,i} \right) = \end{aligned} \tag{19}$$

To generalize the equation, instead of $b_{j,i}$ the average handover rate, λ is used:

$$\sum_{K=L,R} \frac{\sum_{j,MAP_j \in S_k} \left\{ \left[E \left| \frac{1}{3} - \left(\frac{N_j}{N_K} \right) \right| \right] N_K \cdot \lambda \right\}}{N_K} \tag{20}$$

In the M3 model the neighbouring MAPs are grouped into two areas. But it is possible any other grouping as well. For this case equation 20 transformed into fully general form:

$$\sum_{K,\forall G} \frac{\sum_{j,MAP_j\in S_k}\left\{\left[E\left|\frac{1}{n_G+1}-\left(\frac{N_j}{N_K}\right)\right|\right]N_K\cdot\lambda\right\}}{N_K} \qquad (21)$$

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

VII. Simulation results

In this section we compare the accuracy of our Markov movement models to other models found in the literature. The estimation procedure was validated by a simulation environment of a cell cluster shown in Figure 12.a.

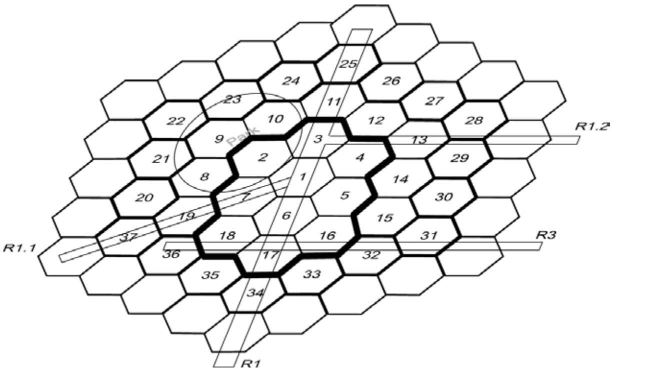


Fig. 12.a The logical cell-cluster in the simulation environment

The simulation was written in the open source OMNet++ using C++ language. 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.

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 parameterized 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). We 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 paper.

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* and *M7* models were used in the simulation as references.

We used the MMCF parameter calculation algorithms, introduced above in Section V for the simulation environment. The smaller examined area contains the cells in the bold circle (cell 1-7, cell 16-18) in Figure 13. In MMCF generated optimal Markov movement model estimation compared to the fix *M3*, *M7* models. The input parameters of MMCF for this simulation environment are:

S={handover during voice call}, ε_c=5, ε_{uv}=0.4, ε_{op}=0.2,

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

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

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

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

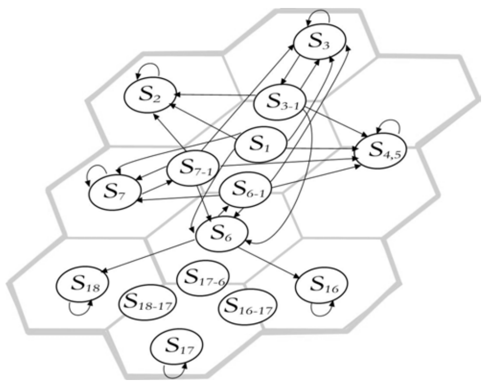


Fig. 13 The TAEV values in RW, ExtRW, M3 and M7 models with direction=(1,4)

The following plots (Figure 14.) 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 Markov approach holds the average error rate, it followed the changes in user motion appropriately, and 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.

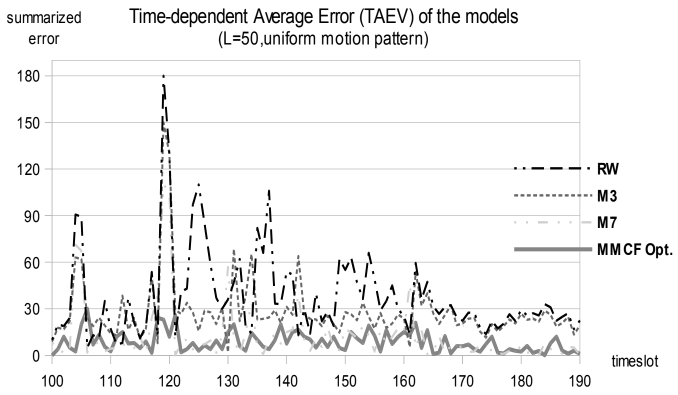


Fig. 14 The TAEV values in RW, M3, M7 and optimal MMCF model

CONCLUSIONS

In this paper we selected some significant parameters of mobility and proposed a method to model the mobile node and the network independently of the technology used. We proposed a simple classification for Markov mobility models, and we have shown examples for the most important types. We showed the attributes of a general Markov model, and we prepared processes for definition. Obviously these algorithms could be further refined.

By using the MMCF 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. The network operator may use this Markov model to make predictions on the future distribution and location of users among radio cells. It is able to support self-configuring system in 4G mobile networks as well.

REFERENCES

[1] Harri Holma (Editor), Antti Toskala (Co-Editor), “WCDMA for UMTS: HSPA Evolution and LTE, 4th Edition”, ISBN: 978-0-470-31933-8, Wiley and Sons, 2007.

[2] László Bokor, Zoltán Faigl, Sándor Imre, "Flat Architectures: Towards Scalable Future Internet Mobility", Lecture Notes in Computer Sciences; LNCS 6656, In: John Domingue, et al. (ed.) Future Internet Assembly 2011: Achievements and Technological Promises. 2011. pp. 35-50.

[3] F. Yu and V. C. M. Leung, “Mobility-based Predictive Call Admission Control and Bandwidth Reservation in Wireless Cellular Networks,” Elsevier Computer Networks, vol. 38, no. 5, Apr. 2002, pp. 577-589.

[4] H. Debar, M. Dacier, and A. Wespi, “A Revised Taxonomy for Intrusion-Detection Systems”, Annales des Telecommunications, vol. 55, 2000, pp. 361-378.

[5] B. Sun, F. Yu, K. Wu and VCM Leung, “Mobility-Based Anomaly. Detection in Cellular Mobile Networks,” ACM WiSe'04, Philadelphia, PA,. 2004, pp. 61-69.

[6] M. Dottling and I. Viering, “Challenges in Mobile Network Operation: Towards Self-Optimizing Networks,” Proc. IEEE Int’l. Conf. Acoustics, Speech, and Sig. Processing, Apr. 2009, pp. 3609–12.

[7] X. Hong, T. Kwon, M. Gerla, D. Gu and G. Pei, “A Mobility Framework for Ad Hoc Wireless Networks”, In Proceedings of ACM 2nd International Conference on Mobile Data Management, (MDM 2001), Hong Kong, (2001, January).

[8] Bo Sun, Fei Yu, Kui Wu, Yang Xiao, Victor C. M. Leung, “Enhancing Security using Mobility-Based Anomaly Detection in Cellular Mobile Networks”, Vehicular Technology, IEEE Transactions on (Volume:55 , Issue: 4) July 2006.

[9] 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.

[10] Sándor Szabó, "The effects of user mobility on the performance of wireless networks", Ph.D dissertation, 2010.

[11] Hanzo, L.; Haas, H.; Imre, S.; O'Brien, D.; Rupp, M.; Gyongyosi, L.: "Wireless Myths, Realities, and Futures: From 3G/4G to Optical and Quantum Wireless", Proceedings of the IEEE, Volume: 100, Issue: Special Centennial Issue, pp. 1853-1888, May 2012.

[12] Wencho Ma, Yuguang Fang, Phone Lin, “Mobility Management Strategy Based on User Mobility Patterns in Wireless Networks”, IEEE Transactions on Vehicular Technology, vol. 56. no. 1., pp 322-330, January 2007.

[13] P. Fülöp, T. Szalka, S. Szabó and S. Imre, “Accurate Mobility Modelling and Location Prediction based on Pattern Analysis of Handover Series in Mobile Networks”, Mobile Information Systems (ISSN: 1574-017X), pp. 1-35, 2009.

[14] Shiang-Chun Lion, Queh-Min Huang, “Trajectory Predictions in Mobile Networks”, International Journal of Information Technology, vol. 11 no. 11 2005.

[15] Wencho Ma, Yuguang Fang, Phone Lin, “Mobility Management Strategy Based on User Mobility Patterns in Wireless Networks”, IEEE Transactions on Vehicular Technology, vol. 56. no. 1., pp 322-330, January 2007.

[16] P. Fülöp, B. Kovács, and S. Imre, “Mobility Management Algorithms for the Client-driven Mobility Frame System – Mobility from a Brand new point of view”, Mobile Information Systems (ISSN: 1574-017X), pp. 313-337, 2009.

[17] P. Fülöp, B. Kovács and S. Imre, “Enhanced Mobility Management Modelling Framework”, Proc. 6th Computer Information Systems and Industrial Management Applications (CISIM 2007), Elk, Poland, June 28-30, 2007. pp. 53-58.

[18] Fang, Y., & Lin, Y., Portable Movement Modelling for PCS Networks, , IEEE Transactions on Vehicular Technology, 1356-1362, 2000.

[19] Ahed Alshanyour and Anjali Agarwal, „Three-Dimensional Markov Chain Model for Performance Analysis of the IEEE 802.11 Distributed Coordination Function” Global Telecommunications Conference, GLOBECOM 2009.

[20] M. M. Zonoozi and P. Dassanayake, "User mobility modeling and characterization of mobility patterns", IEEE Journal on Selected Areas in Communications, Volume 15, No.7, September 1997.

[21] Y.-B. Lin, M. Chen, and H. Rao, “Potential Fraudulent Usage in Mobile Telecommunications Networks”, IEEE Transactions on Mobile Computing, vol. 1, no. 2, 2002, pp. 123-131.

[22] P. Fazekas, S. Imre, M. Telek: ”Modeling and Analysis of Broadband Cellular Networks with Multimedia Connections”, Telecommunication Systems (Kluwer Academic Publishers) Vol. 19, No. 3-4, pp. 263-288, March-April 2002 .

[23] Shiang-Chun Lion, Queh-Min Huang, “Trajectory Predictions in Mobile Networks”, International Journal of Information Technology, vol. 11 no. 11 2005.

Péter Fülöp received his M.Sc. degree in Technical Informatics from the Budapest University of Technology and Economics in 2005. He is a former PhD student on the faculty of Informatics on the Department of Telecommunication and fellow researcher in the Mobile Innovation Centre of BME. Currently he is working as a system test engineer at Ericsson Telecommunications Hungary at Research and Development Department. His research interests are IP mobility, interworking of heterogeneous mobile networks and movement modeling in cellular, mobile networks. The PhD thesis is going to be written on Complex Mobility Management Systems and Applications.

Sándor Szabó received the M.Sc. degree in Electronic Engineering from the Budapest University of Technology (BME) in 2000. Next he started his Ph. D. studies at BME. He received the Ph.D. degree in 2011. Currently he is carrying his teaching activities as an assistant professor at the Dept. of Telecommunications of BME. He leads research projects, and also a project leader at the Mobile Innovation Centre of BME. His research interests include mobile and wireless systems, mobility modeling, mobility management and the integration of heterogeneous wired and wireless networks.