Predictive Mobility Models based on Kth Markov Models

F. Lassabe, P. Canalda, P. Chatonnay, F. Spies
LIFC - Laboratoire d'Informatique de l'Université de Franche-Comté
Numérica - Multimedia Developpement Center
Cours Louis Leprince Ringuet, BP 21126
25201 Montbéliard Cedex, France
Email: {frederic.lassabe,philippe.canalda,pascal.chatonnay,francois.spies}@pu-pm.univ-fcomte.fr
D. Charlet
INRIA Rocquencourt
Email: damien.charlet@inria.fr

Abstract

With the massive arrival of wireless networks, the mobility of the terminals increases with the interconnections. New problems, such as mobile multimedia content streaming, arise with the emergence of new mobile multimedia services. In this paper, we present a mobility model adapted to the logging of mobile positioning or the tracking of mobiles. This model is based on the Markov Models, especially the All-Kth Markov Model. We present two predictive models from the AKMM: the K-to-1 past model and its improvement, the K-to-1 past* model. Both are pertinent solutions to tackle mobility patterns. We validate our approach firstly with various realistic benchmarks on data related to indoor WiFi positioning systems. Lastly our approach is suitable to experimental projects, such as GuiNuMo, addressing the service continuity in the scope of mobile multimedia services.

Keywords: *Continuous Multimedia Services, Mobility patterns, WiFi Indoor Positioning, Handoff, Learning System, Markov Models.*

I. Introduction

With the spreading of wireless networks and their associated services, new problems arise. In particular, the continuity of the services provided or the routing must be ensured in the mobility. An approach to address the service continuity is the handoff anticipation. The mobility prediction is a potential technique to anticipate the handoff. It requires the positioning of the mobile terminal.

Figure 1 describes the step decomposition of the mobility prediction process. First, the mobile is either positioned or tracked, which requires positioning algorithms. Second, the result of the mobile positioning and tracking is logged for further analysis. Third, the logs are exploited configuring or training a mobility prediction model. And fourth, based on the data learned, the mobility prediction system is able to predict mobiles' movements using their previous ones, to compare such movements to the prediction model.

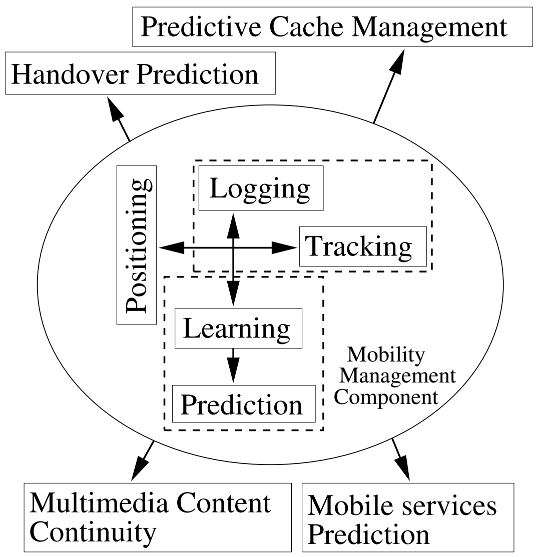


Fig. 1. Mobility prediction steps.

Our work particularly focuses on WiFi technology [1], because of its low cost and high bandwidth capability. Thus, it comes within the scope of multimedia content streaming, which requires service continuity and adaptability of the network to the clients' mobility.

In this article, we present Markov Model-based mobility models. These aim at predicting middle-term movements. Moreover, the WiFi positioning system, although being accurate, still produces errors. In such a context, trajectory computation models prove to be unsatisfying. Thus we base our mobility models on Kth Markov Models, especially the All-Kth Markov Model, which are learning models. These mobility models aim at taking into account yet simple mobility patterns, but which allow predicting next cell crossing.

First, we present the work related to positioning systems and mobility prediction. Then, we expose our contributions to the mobility prediction. Finally, we analyse real benchmarks experiments, we conclude and present future trends.

II. Related Work

The mobility management in a wireless networks requires to address two problems. Firstly, the mobile terminals must be located during a phase called positioning. Secondly, modeling their movements allows to predict their future locations and acting according to it.

A. Mobile terminal positioning

Mobile terminal positioning is the first step of mobility management process. Predicting the mobile terminal movements requires modeling the movements of the mobiles. Several projects have been initiated in the scope of WiFi indoor positioning. Whether using a reference points-based approach, like the RADAR system [2] or a radio wave propagation model [3] [4], these models use signal strength measurements in order to achieve the positioning. After studying and testing these models, we found it was not fully adapted to highly heterogeneous environments such as our testbed. Therefore, we developed a model based on the hybridation of the Friis based calibrated model (FBCM), presented in [5], and the RADAR-like approach. We call it the FBCM and RADAR-like based Hybrid Model (FRBHM).

B. Various models of mobility prediction

The mobility prediction can be achieved by trajectory computation [6], allowing to predict short-term movements based on the speed vector of an object. As WiFi positioning is not sufficiently robust, various methods allow to smooth the mobile's trajectory, such as Kalman filters [7] and double exponential smoothing [8]. However, all of these methods require fine-grained positioning to be used. Therefore, they are mostly suited to virtual reality devices than to WiFi mobile positioning.

In [9], J. Pitkow et al. describe how to mine the *longest repeating sequences* (LRS) in users paths. The method consists in identifying in users path the subsequences which follow these criteria:

- subsequences are composed of a set of consecutive items,
- they must be repeated T times, with T typically equal to 1,
- at least once, the LRS is the longest repeating (ie. it is not part of a longest LRS).

A refaire

In [10], the authors extend the use of the All-Kth Markov Model by pruning some states to reduce the size of the automata. Their model's name is *Selective Markov Model* (SMM). The states are pruned when they do not carry much sense. A state has not much sense either when occuring not frequently or when having few difference between the probabilities on its outgoing transitions or when having a bad error rate during the validation step. The SMM is an approach similar to the LRS one.

In [11], the authors present a hierarchical mobility model. The local mobility model uses trajectory computation to make short-term prediction whereas the global mobility model considers user mobility patterns (UMP) and makes middle-term prediction by matching the UMP with the user's actual path (UAP). Merging those models allows to determine next cell crossing with accuracy and increases the matching of the UAP with the UMPs.

Another project [12] uses an *Hidden Markov Model* (HMM) to determine the mobile's future position after a learning phase. An HMM is a double stochastic model. Its use takes into account the errors in the observation of the mobile terminals' location. Indeed, in most of the indoor positioning systems, the accuracy is such as the observation can not be considered as real. The observations are matched against the reality according to a stochastic process. The probabilities of the transitions from one state to another compose the second stochastic process.

The tutorial [13] of L. R. Rabiner defines an HMM as being composed of

- a set of physical states,
- a set of observation symbols to be matched with the set of physical states,
- A, the state transition probability distribution,
- B, the observation symbol probability distribution whitch matches the observation symbols with the physical states,
- π, the initial state distribution.

The compact notation for the HMM is $\lambda = (A, B, \pi)$. The problem to be solved with the HMM and when predicting the next movement of a mobile terminal is: given the observation sequence $O_1 O_2 ... O_T$, with O_i in the set of observation symbols, and a model λ, how do we select a corresponding state sequence $q_1 q_2 ... q_T$, with q_i in the set of physical states, which is optimal according to the current problem's criteria ? Using a Viterbi algorithm, the problem can be solved.

Due to the accuracy and refresh time of indoor positioning systems, the trajectory computation is not possible. The HMM approach is interesting, but should take into account more than 1 previous state. The Selective Markov Model is interesting too but should consider time to provide real-time services such as multimedia content streaming and handoff.

III. Contributions

As a pilar of our following contributions, we have tested the FRBHM in [14]. We obtained a mean accuracy close to the meter with a standard deviation of 30 centimeters. It is sufficient to consider location-based services, such as the mobility prediction within a WiFi indoor multimedia guide. We are interested in middle-term prediction, based on the observation of the physical locations of the mobile terminals. As trajectory computation is not suited to middle-term prediction, we consider a discrete representation of the physical locations. For example, a physical location could be a base station's coverage area.

Initial works in the field of web pages access prediction [9] are suited to the modeling of mobility prediction (MP). We base our work on *All-Kth Markov Model* (AKMM). In the scope of the service continuity, we address in particular the multimedia content streaming. Our work is connected with the cache sibling as presented in [15].

We consider the predictive mobility of pervasive terminals when trajectories are complex and for various usages among positioning, handoff policy and multimedia cache management. Hence by fistly considering, and secondly anticipating, the trajectory direction of mobile terminals, we improve the service continuity.

In the sequel, we briefly introduce the Kth Markov Model. Then we present our improvements, based on the AKMM.

A. The Kth Markov Model (KMM)

We consider the Kth Markov Model (KMM) as a model representing K past states (geographical locations) in one KMM state. For N defined geographical locations, the KMM contains at most $N \times (N-1)^{K-1}$ states.

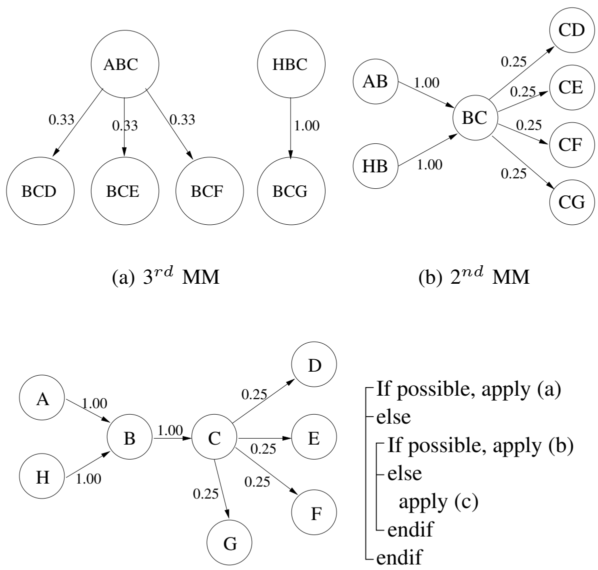
The KMM is defined as follows:

- S is the set of states,
- T the state transition probability distribution,
- S$_i$ the set of the initial states of the system.

In our model, S$_i$ contains one state: the *out* state. It represents the state of a mobile not in the system. When a mobile is detected in the system for the first time, it is considered having just followed the transition from the out state to another state of the system. There are two ways for a mobile to go from the system to the out state. It can request to quit the system. The connection also can be lost, therefore leading the system to consider the mobile out of it.

Among the steps identified in the mobility prediction process, only the training is offline. It consists in feeding data to the system. For each peregrination, the logical states corresponding to K geographical states are built and the transitions are labeled with the probability for a mobile to go through the transition. The training is the step in which the set T is built. The training step is offline, but online training can be performed, in particular to update the model when its accuracy decreases.

Considering the following peregrinations: ABCD, ABCE, ABCF and HBCG, with each character being a location, the 3rd MM, 2nd MM and 1st MM representations are depicted in figure 2.



(a) 3rd MM

(b) 2nd MM

(c) 1st MM

If possible, apply (a)
else
  If possible, apply (b)
  else
    apply (c)
  endif
endif

(d) All-3rd MM algorithm

Fig. 2. The All-3-th Markov Model Sample.

B. The K-to-1 past Model

The AKMM is similar to the KMM but it builds the automata for K to 1 past geographical states. Considering the same example as in the section III-A, the All 3^{rd} Markov Model would merge the 3 automata of figure 2. The KMM's drawback is not taking into account more than one size of past user's path. The K-to-1 past model is based on the AKMM in which we add a probability threshold. When predicting the next movements, we do not select the most likely to occur, but we select every next state for which the probability to occur is greater than the threshold. Several states are selected and thus, the accuracy is improved.

The prediction is performed by constructing the logical state corresponding to a succession of geographical states. A search in the automata from degree K to 1 determines the next potential states by comparing their probability to occur to a probability threshold. Actually, the prediction in a *All-K-th Markov Model* works as follows (see fig. 2(d)): beginning by the highest Markov degree automaton, the algorithm determines potential transitions for which the probability to occur is greater or equal to the threshold. If there is one or more solution, the algorithm returns a set composed of these solutions. If there is no transition satisfying to the threshold, the algorithm determines the transitions on the next automaton. The next automaton is the automaton which degree is 1 below the automaton's degree tested. The algorithm continues testing automata until either it finds solutions for a degree or he has tried 1-st Markov Model and has found no solution. The threshold represents the minimum probability for preparing for the handoff.

We call *K-to-1 past* the AKMM with probability threshold. The probability threshold can trigger two actions :

- prefetching multimedia content,
- planning a handoff.

The goal of cache prefetching [15] is to increase the success rate of handoff cache policies while not congestioning the network by prefetching each cache in the neighbourhood.

C. The K-to-1-Past* Model

Mobility patterns are of various sizes and a user could follow partially a pattern. It implies that predicting with the highest order's Markov Model could lead to an error. To address this, we agregate the predictions given by the K-th Markov Model to the 1-st Markov Model. We build the automata from the K-th order to the first order, like in the K-to-1 past model. Then, the prediction is composed of the union of the prediction's sets for the Kth to the first order Markov Model. Thus, increasing the model's order increases the accuracy. Given R_i the resulting set of the prediction for the ith Markov Model, the result for the K-to-1-Past* Model is

$$R_{K-to-1-past^*} = \bigcup_{i=1}^{K} R_i$$

IV. Experiments

Experiments have been carried out in order to validate our propositions. First, we test the Hybrid Model, then the K-to-1 past prediction model. Finally, we present the architecture of the mobility management middleware.

A. The hybrid model

Table I exposes the overall results for different major models we have tested. That reveals the indoor positioning's accuracy brought by the FRBHM. Such results prove the feasability of WiFi-based indoor location system. It also renders pertinent the study of mobility management based on the mobile terminals' positioning.

	Mean error (m)	Standard deviation
Interlink Net.	29.38	12.17
SNAP-WPS	22.78	14.07
FBCM alone	15.86	9.34
FRBHM	1.07	0.32

TABLE I. Precision of the positioning models.

B. Implementation

We have chosen to use LASH [16] to implement our mobility prediction model. Indeed, the Markov Models generated by the training are stored as automata. The mobile terminals' movements are represented in log files by four fields for each movement: the source location, the destination location, the duration in seconds and the unix date. The duration is computed as being the difference between the date when the mobile terminal entered the destination location and the date it entered the source location. The duration and the date are not used now but we log it for further work. The source and destination location are mandatory to log the movements.

In table II, we give an example of log file. Each line describes a transition. The first entry is the source state, the second entry is the destination state. *Away* is a generic state meaning that the mobile is out of the mobility prediction model. The third entry is the duration in seconds the mobile stayed in source state and the fourth entry is the unix date at which the transition occured.

A generic state *away* is either representing a mobile terminal out of the system or a mobile terminal which has disconnected, for example because it reached the end of

Source	Destination	Duration	Date
Away	Corridor	0	0
Corridor	Office 01	9	9
Office 01	Corridor	120	129
Corridor	Meeting Room	34	163
Meeting Room	Corridor	3584	3747
Corridor	Away	13	3760

TABLE II. Structure of a log file.

its autonomy. A movement is called a *transition*. A set of transitions beginning and ending in the generic state away is called a peregrination.

In order to test the model, the program uses 75 % of the input data to train the model and the remaining 25 % to validate its precision. By 75/25, we mean peregrination rather than transition.

C. The K-past model

We have tested the K-to-1 past model with 2 context databases provided by the **Institut fur Pervasive Computing**[1]. One is the *Augsburg Indoor Location Tracking Benchmarks* and the other is the *Nokia Context Data*. The Nokia context data records the GSM status of a user. The Augsburg Location Tracking is composed of two series of log files. The *_fall were built during the autumn, and the *_summer were built during the summer. The prefix a, b, c and d identify 4 distinct users.

Log file	(learning/validation percent)	1-past
a_fall	(75/25)	0.7841
b_fall	(75/25)	0.8131
c_fall	(75/25)	0.6592
d_fall	(75/25)	0.7127
a_summer	(75/25)	0.5000
b_summer	(75/25)	0.7857
c_summer	(75/25)	0.7375
d_summer	(75/25)	0.3333
Overall	a_fall to c_fall (100/0) d_fall (94/6), *_summer (0/100)	0.6470

TABLE III. Mobility prediction accuracy, 1-past, Augsburg files.

Table III presents the first test on the Augsburg Location Tracking. It consists in separated tests for each file in the package and a test on the overall. The overall test is carried out by concatenating all the files together. For each test, the training is made considering the first 75% peregrinations of the log file. The remaining 25% are used to test the training. The prediction is realized on the transition with the biggest weight. Only one transition is selected, like it is selected in [10] and [9]. Table III shows a good accuracy for most of the files. We find only 2 files whose accuracy is less or equal to 50%. These results are explained by the size of the log file, which contains few peregrinations. Thus, the training is not exhaustive, leading the tests to wrong predictions.

Globally, the results are good because the training is made on the same user's profile than the test. It proves the influence of the context within the scope of the mobility prediction. The lower accuracy observed in the overall test is logical. Indeed, the concatenation of the files into another one involves that the system uses the 4 user's fall profiles to train before confronting the resulting model to the user's movements in summer. As the users may have different habits over different periods, the accuracy is lower.

Threshold	1-past	2-past	3-past	4-past
1.0	0.0694	**0.2729**	0.2705	0.2077
0.9	0.1632	**0.3551**	0.3430	0.2754
0.8	0.2882	**0.4589**	0.4324	0.3527
0.7	0.2899	**0.5459**	0.4831	0.3792
0.6	0.3368	**0.5676**	0.4976	0.3816
0.5	0.5191	**0.6618**	0.5507	0.4082
0.4	0.6458	**0.6667**	0.5556	0.4082
0.3	**0.7396**	0.7198	0.5749	0.4251
0.2	**0.7951**	0.7560	0.5942	0.4251
0.1	**0.9080**	0.8213	0.6135	0.4324
0.0	**0.9583**	0.8430	0.6256	0.4444

TABLE IV. K-past Mobility prediction accuracy, Nokia Context Data.

Threshold	1-past	2-past	3-past	4-past	10-past
1.0	0.1719	0.1255	**0.2937**	0.2862	0.0688
0.9	0.4758	**0.4814**	0.4526	0.4359	0.1078
0.8	**0.4879**	0.4814	0.4563	0.4638	0.1143
0.7	0.4879	0.4814	0.4563	**0.4926**	0.1217
0.6	0.4879	0.4814	0.4563	**0.5344**	0.1236
0.5	0.4879	0.4851	0.4600	**0.5530**	0.1320
0.4	0.4879	0.5279	0.5074	**0.5967**	0.1366
0.3	0.4879	0.6190	0.5864	**0.6478**	0.1561
0.2	0.6097	0.6887	0.6571	**0.7026**	0.1757
0.1	**0.8457**	0.8429	0.8123	0.7667	0.1914
0.0	**0.9898**	0.9665	0.9210	0.8411	0.1961

TABLE V. K-past Mobility prediction accuracy, Augsburg Location Tracking.

Table IV and table V expose the accuracy on, respectively, the Nokia Context Data and the Augsburg Location Tracking when using a Kth Markov Model. We draw several conclusions from these tests:

- first, the lower the probability threshold, the better the accuracy is. The reason is that the decrease of the threshold leads to an increase of the number of neighbour states prepared for the handoff, which implies a greater probability of success,
- second, the 1-past accuracy is better than the other's ones below a given threshold. This is explained be-

[1] http://www.pervasive.jku.at/Research/Context_Database/index.php

cause the 1-past training gives more possible transitions (in fact, every transition occuring in the training data set) from one state. Thus, the transitions have less probabilities to occur than the transitions in the 2+ past automata. It explains that the 1-past has a lower accuracy with high thresholds but becomes better with low thresholds,

- third and concerning the $K > 1$ past, the accuracy decreases while K increases. This is due to the increased size of the path stored in each state of an automaton. Indeed, the longer the past, the lower is the probability to observe the same path in further peregrinations, thus reducing the overall accuracy of the model. However, this analyse is not absolutly exact. There could be some cases whose precision would grow with the increase of the automaton's degree, for example in an environment which users always follow long paths crossing from time to time,
- fourth, for several thresholds, the accuracy remains the same and then suddenly increases. It is due to the consideration of a transition frequently taken. Thus, such a transition has a high probability to occur, whereas the other have a small probability. This explains that the success of the prediction suddenly grows for a low threshold, while it chosed only one possible state for higher thresholds. In the Augsburg's data set, which represents movements in an office building, that corresponds to a user who often goes to his own office while he goes rarely in other places. It leads to a bad accuracy, because of the number of low probabilities. In table V, we see that one destination overcomes the other for a threshold between 0.8 and 0.3. But it represents only 48% of the overall outgoing transitions from the states of the automaton.

Based on the KMM, the K-past model produces interesting results. The 1-past becomes the most accurate below a threshold, when it selects more transitions then the other degrees. When considering higher threshold, the accuracy increases until a degree often greater than 1 and then decreases. Lowering the accuracy when taking into account longer paths is not the goal of a mobility prediction model. That is why we conceived and put to test the K-to-1 past model.

D. The K-to-1 past model

Table VI gives results on the Nokia Context Data for the K-to-1 past model. We observe that increasing the K-to-1 past degree does not imply an increase of the accuracy. It is not surprising in the actual state of the K-to-1 model and we give the following explanation: let use again the figure 2. Let the threshold be 0.2. Now let us consider a new peregrination, ABC, then G. We expect to predict the last position of the mobile, given its past peregrination ABC. When this peregrination is given to the prediction algorithm, it begins trying to find potential transitions in the 3-rd Markov Model. In the current case, there is a third degree state ABC. This state has three potential transitions, each occuring with a probability of 0.33. Thus, the algorithm returns every three target states: BCD, BCE and BCF. It means that the mobile is predicted to go either to state D, E or F. But the mobile goes in G, so the prediction is wrong. If the prediction is made with the 2-nd degree automaton, the physical states returned are D, E, F and G, which is more convenient.

Threshold	1-past	2-past	2-to-1 past	3-past	3-to-1 past
1.0	0.0694	0.2729	0.2923	0.2705	**0.3357**
0.9	0.1632	0.3551	0.3744	0.3430	**0.4082**
0.8	0.2882	0.4589	0.4855	0.4324	**0.5097**
0.7	0.2899	0.5459	**0.5725**	0.4831	0.5652
0.6	0.3368	0.5676	**0.5894**	0.4976	0.5797
0.5	0.5191	0.6618	**0.7029**	0.5507	0.6667
0.4	0.6458	0.6667	**0.7150**	0.5556	0.6715
0.3	0.7396	0.7198	**0.7681**	0.5749	0.7005
0.2	0.7951	0.7560	**0.8092**	0.5942	0.7303
0.1	0.9080	0.8213	**0.8816**	0.6135	0.7778
0.0	**0.9583**	0.8430	0.9203	0.6256	0.7923

TABLE VI. Mobility prediction accuracy with K-to-1 past, Nokia Context Data.

Finally, we observe that the K-to-1 past is better than the K-past alone. It is obvious as the next transition is searched amongst several automata.

E. The K-to-1-past* Model

The last test presented in this paper concerns the K-to-1 past*. As described in the contributions, this model returns the union of the set issued from each K-past model's degree. Again, the tests were run on the Augsburg data files and the Nokia Context Data.

Threshold	1-past	2-to-1 past	3-to-1 past	12-to-1 past
1.0	0.0694	0.2483	0.3142	**0.5017**
0.9	0.1632	0.3663	0.4080	**0.5590**
0.8	0.2882	0.5243	0.5503	**0.6545**
0.7	0.2899	0.5660	0.6042	**0.6875**
0.6	0.3368	0.6441	0.6632	**0.7222**
0.5	0.5191	0.7066	0.7274	**0.7882**
0.4	0.6458	0.7535	0.7830	**0.8229**
0.3	0.7396	0.8299	0.8490	**0.8715**
0.2	0.7951	0.8628	0.8802	**0.8906**
0.1	0.9080	0.9288	0.9288	**0.9358**
0.0	**0.9583**	**0.9583**	**0.9583**	**0.9583**

TABLE VII. Mobility prediction accuracy with agregated K-to-1 past, Nokia Context Data.

Tables VII and VIII show that the accuracy grows with the model's order. Indeed, by agregating the predictions

Threshold	1-past	2-to-1 past	3-to-1 past
1.0	0.1719	0.1766	**0.3615**
0.9	0.4758	0.4879	**0.4879**
0.8	0.4879	0.4879	**0.4879**
0.7	0.4879	0.4879	**0.4879**
0.6	0.4879	0.4879	**0.4879**
0.5	0.4879	0.4916	**0.4916**
0.4	0.4879	0.5344	**0.5409**
0.3	0.4879	0.6255	**0.6255**
0.2	0.6097	0.7054	**0.7082**
0.1	0.8457	0.8866	**0.8931**
0.0	**0.9898**	**0.9898**	**0.9898**

TABLE VIII. Mobility prediction accuracy with agregated K-to-1 past, Augsburg Location Tracking.

from the Kth order to the first order, we increase the accuracy of the model of approximately 14 % for the 3-to-1 past.

We also notice two main phenomena: first, the accuracy is the best for the zero-threshold. It is expected as the zero-threshold makes a prediction with every transition outgoing of the current state. What is interesting is that the accuracy for the zero-threshold is the same for the first degree and the higher ones. It is explained by the composition of the first degree model. In fact, it contains every possible transition from one state to another. Thus, every transition of a higher degree Markov model exists in the first degree model. The only difference is that the higher degrees models do not contain every transition from the first degree model, relatively to a given succession of states, so their probabilities are higher. For a threshold equal to zero, the first degree model can make a prediction as accurate as the higher degrees models' one.

Second, due to the same properties of the model, the accuracy increases better for a high threshold. When the threshold is 1, the first degree model has a very poor accuracy whereas the increase by using the second degree and third degree models is important.

V. Conclusions

In this paper we have presented mobility prediction models wellsuited to WiFi indoor positioning and multimedia system. WiFi is chosen because of its accessibility the public and its high bandwidth for data transfer. The mobility prediction consists in four main parts: the positioning of the mobile terminals, the logging of its locations, the training of the mobility model according to the logs and the real-time mobility prediction.

We have shown [14] that our positioning system, the FRBHM, is efficient and can be used in mobility management, particularly within the scope of mobility prediction. The logging requires to know what is needed to achieve the prediction. We chose a simple format which remains sufficient to provide learning of the simple mobility patterns.

Our *K-to-1 past* model carries out the mobility prediction. It combines the *All-K-th Markov Model* with a transition threshold to select the most probable future states. Its precision is studied in details and we consider that its accuracy is correct. Its improvement as K-to-1 past* is even more accurate. However, other improvements can be brought to this model.

We have observed some properties of the K-to-1 past* model applied to the mobility prediction. While it is adapted to the mobility prediction with repeating mobility patterns, it cannot exploit others patterns such as periodic patterns. Periodic patterns are repeating at precise times, for example, the mobility patterns in a mall are repeating differently according to the day of week.

The applications of mobile positioning and mobility prediction are numerous. Mobile digital guides, for example, are applications very dependent on the positioning. The mobility prediction particularly applies to handoff anticipation. The latter ensures continuity of the services with more reliability. In particular, our mobility management system is used in the MoVie project [17]. The GeoMoVie component manages mobility and allows the continuity of the streaming of multi-media flows. The mobile tracking and the prediction also has great potential witin the scope of wireless networks security.

VI. Future work

Concerning the positioning of the mobile terminals, the accuracy is sufficient to be used in various applications. However, we must determine the limits of this model. Although its accuracy is good, we cannot use it in instant trajectory computation. Therefore, the mobility prediction remains possible at a high level (BTS, cell, etc.) but it stays uncertain to determine a mobile's future location in very short time.

The mobility prediction needs more improvements to be fully functionnal. First, the multimedia streaming and the handoff preparation are real-time procedures. Thus we should consider the time in the *K-to-1 past* model. We plan to label the transitions not only with their probability to occur but also with statistical data on the time taken to follow the transitions by the mobiles of the training set.

Second, it is not optimal that the increase of the accuracy requires to select much states for the prediction. Indeed, if applied to multimedia content prefetching, selecting many caches to be preloaded increases most certainly the accuracy of the prediction but it generates much traffic on the network. We think about combining the K-to-1 past* with a quality level in order to prefetch part of the data requested. Thus, the accuracy should lower a bit but the network would not be overloaded. The partial prefetching also allows to preload more caches than considering only full or no prefetch. The improvements described above are short-term improvements.

The middle-term improvements concern the study of mobility patterns. We consider using time and population-based patterns. The first problem is to identify the patterns. The population-based patterns require to associate each peregrination to a population. The population is a set of peregrinations having common points. When the populations are defined, we think about using an *Hidden Markov Model* to make predictions according the various populations defined. Time-based patterns require studying the sequences repeated over time. Identifying either time-based patterns or populations of users will improve the prediction's accuracy.

Handoff is still in development state. It is necessary to specify the protocol used to communicate in unconnected mode. We foresee how to manage power saving in the communication protocol. In addition, the mobility prediction and a handoff protocol can be included in any mobile network.

References

[1] B.P. Crow, I. Widjaja, J.G. Kim, and P. Sakai. IEEE 802.11 Wireless Local Area Networks. *IEEE Communications Magazine*, 35(9):116–126, September 1997.

[2] P. Bahl and V. N. Padmanabhan. RADAR: An in-building RF-based user location and tracking system. In *INFOCOM (2)*, pages 775–784, 2000.

[3] Interlink Networks, Inc. A practical approach to identifying and tracking unauthorized 802.11 cards and access points. Technical report, 2002.

[4] Y. Wang, X. Jia, and H.K Lee. An indoors wireless positioning system based on wireless local area network infrastructure. In *6th Int. Symp. on Satellite Navigation Technology Including Mobile Positioning & Location Services*, number paper 54, Melbourne, July 2003. CD-ROM proc.

[5] F. Lassabe, D. Charlet, P. Canalda, P. Chatonnay, and F. Spies. Friis and Iterative Trilateration Based WiFi Devices Tracking. In *Proceedings of the 14th Euromicro Conf. on Parallel, Distributed and Network-based Processing (PDP 2006)*, Februar 2006. to be published.

[6] S.J. Lee, W. Su, and M. Gerla. Mobility prediction in wireless networks. In *Proceedings of IEEE ICCCN*, pages 22–25, October 2000.

[7] A. Kiruluta, M. Eizenman, and S. Pasupathy. Predictive head movement tracking using a kalman filter. *IEEE Transactions on Systems, Man and Cybernetics - Part B: Cybernetics*, 27(2):326–331, April 1997.

[8] Joseph J. LaViola Jr. Double exponential smoothing: An alternative to kalman filter-based predictive tracking. In *Proceedings of the workshop on Virtual environments 2003*, pages 199–206. ACM Press, 2003.

[9] J. Pitkow and P. Pirolli. Mining longest repeating subsequences to predict world wide web surfing. In *Proceedings of USITS'99: The 2nd USENIX Symposium on Internet Technologies and Systems*. USENIX, USENIX, October 1999.

[10] M. Deshpande and G. Karypis. Selective markov models for predicting web page accesses. *ACM Transactions on Internet Technology (TOIT)*, 4(2):163–184, May 2004.

[11] T. Liu, P. Bahl, and I. Chlamtac. Mobility modeling, location tracking and trajectory prediction in wireless atm networks. *IEEE Journal on Selected Areas in Communications*, 16(6):922–936, August 1998.

[12] J.M. Francois, G. Leduc, and S. Martin. Learning movement patterns in mobile networks : a generic method. In *European Wireless 2004*, pages 128–134, February 2004.

[13] L. R. Rabiner. *Readings in speech recognition*, chapter A tutorial on hidden Markov models and selected applications in speech recognition, pages 267 – 296. Morgan Kaufmann Publishers Inc., 1990.

[14] F. Lassabe, P. Canalda, D. Charlet, P. Chatonnay, and F. Spies. Refining wifi indoor positioning renders pertinent deploying location-based multimedia guide. In *Proceedings of the IEEE International Workshop on Pervasive Computing and Ad Hoc Communications (PCAC06)*, April 2006. to be published.

[15] D. Charlet, P. Chatonnay, and F. Spies. Hand-over video cache policy for mobile users. In J.B. Stefani, I. Demeure, and D. Hagimont, editors, *Proceedings of 6th IFIP International Conference on Distributed Applications and Interoperable Systems (DAIS 03)*, volume LNCS 2893, pages 176–186, 2003.

[16] B. Boigelot. The lige automata-based symbolic handler (lash). Web site. http://www.montefiore.ulg.ac.be/ boigelot/research/lash/.

[17] J. Bourgeois, E. Mory, and F. Spies. Video transmission adaptation on mobile devices. *Journal of Systems Architecture*, 49:475–484, 2003.