Positioning Awareness: an Essential Component for Mobile Multimedia Applications

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

The spreading of the WiFi networks allows new applications. New problems bound to the mobility of the terminals arise. In this article, we deal with the mobil terminal positioning. The positioning service is integrated in a mobility management middleware. The solution proposed is trilateration for which the distances are computed according to the signal strength. We also propose some method to refine positioning in order to increase the precision. The positioning accuracy is evaluated by a large set of tests. Mobil terminal positioning is the first step to context awareness. Then we introduce prediction using Hidden Markov Model and how we use the past to determine the futur. Finaly we show how we use positioning and prediction in two multimedia applications

Keywords*: Mobility Management Middleware, WiFi Positioning, Handover, Trilateration, Friis-based calibrated model, Prediction, Hidden Markov Model.*

I. Introduction

We call position awarness the knowing of the position and the probable futur position of a mobil terminal. Both of these are really usefull in the context of the spreading of wireless networks and their associated services. In particular, the continuity of multimedia services provided must be ensured in mobility. The mobility prediction is a potential technique to anticipate problems arising when a mobil terminal moves from one antenna to another (ie during the handover). Position could also leed to the definition of new services.

Wireless networks are of various types: GSM, UMTS, WiFi [1], etc. The services provided are also numerous, from consulting web pages to watching on-demand video sequences. Because of these facts, it is natural to consider using a middleware to provide the mobility management. This way, the interface between the user's applications and the service continuity component is transparent.

In this article, we present an indoor WiFi positioning and prediction system which is part of a service which aims at ensuring service continuity. this service relay on multiple components: a system learning the mobile terminal moves and a system using the data acquired to anticipate the service interruptions. The handover is managed by a protocol dedicated to the mobility. In this article we focus on two step : the positioning and the prediction.

The service presented comes within the scope of a streaming platform of multimedia content. This project name is MoVie [2]. It is composed of 4 modules (fig. 1):

- NetMoVie integrates the RTP/RTCP protocol. It receives a few video sequence qualities and selects the most adapted one depending on the current situation.
- SysMoVie gathers ORB components and integrates the hierarchy of video caches. The strategy of video cache management is specific to the particular temporal data.
- WebMoVie represents a query interface of the MoVie platform. It is the entry point of clients where they are identified. A trader is used for each query in order

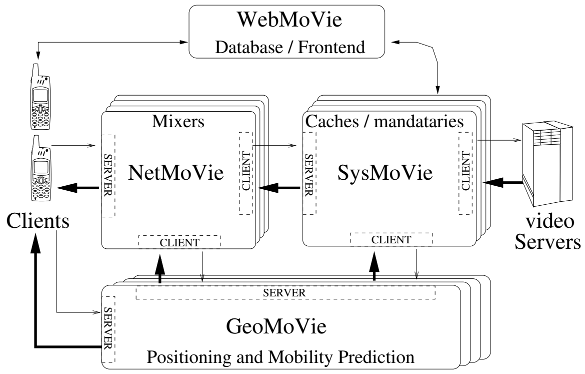

Fig. 1. MoVie structure.

to localize one or more required video sequences in SysMoVie.

- GeoMoVie [3] tracks the mobile clients and anticipates their future moves. It contains positioning and prediction modules. It also provides the handover management.

First, we present the work related to positioning systems and mobility prediction. Then, we expose our contributions to these points. Finally, we describe the experiments for which the results are analyzed.

II. WiFi positioning in the litterature

In the RADAR project [4], a signal strength map is used to position the mobile terminals. A database containing points with known coordinates is built. For each point in the database, the geographical coordinates and the signal strength measured from each access point are stored. The signal strength map is established either by measurements or by computation following a radio wave propagation model. Positioning a mobile requires to measure the signal strength at an unknown point. By comparing the measurement with the ones in the database, one can deduce its position. The error median of RADAR's accuracy is between 2 and 3 meters.

Other projects use measurement sets to position mobile terminals. For example, statistical approaches, [5] and [6], use signal strength distribution on reference points to locate a mobile terminal. A project used the neural network approach [7] to determine a mobile terminal location. These projects are greatly inspired by the RADAR project.

Another way to determine a mobile's position is trilateration. It determines the position of a point, whose coordinates are unknown, by using the distances to reference points. The reference points are points whose position is known at each moment. In the SNAP-WPS project [8], measurements are used in a third degree polynomial regression. The resulting polynomial expression is used to compute the distances corresponding to each signal strength measured. Interlink Networks' (IN) approach [9] uses an alternative to the Friis equation [10] to take the indoor obstacles into account. The accuracy of the SNAP-WPS is about 1 to 3 meters. The system of Interlink Networks has an accuracy of about 3 meters. However, points exist, where the precision is bad. Building topology heterogeneity explain these points. In [11], we present the *Friis-based Calibrated Model* (FBCM) as an improvement of the Interlink Networks' approach.

The RADAR technique is the best in accuracy but its setup time cost is extensive. The IN's approach is quick to set up, but is the less precise. SNAP-WPS is between RADAR and IN in both setup time and precision. The results obtained with the FBCM are not sufficient to provide location-based services. We need a precision of less than 3 meters to consider such services.

Mobile terminal positioning is the first step to mobility management. Predicting the mobile terminal movements requires modeling the movements of the mobiles. The mobility prediction can be achieved by trajectory computation, like in [12], but this requires great precision and refresh rate in the positioning process. Another project [13] uses an *Hidden Markov Model* (HMM) to determine the mobile's future position after a learning phase. In [14], the authors establish a Markov Model with various degrees of learning according to the Kth Markov Model presented in [15]. Then, some states are pruned to reduce the size of the automata and the overall accuracy. Their model's name is *Selective Markov Model* (SMM).

III. Contributions

This part deals with the flaws in the current work related to mobility management. First, we expose our contributions to the mobile terminal positioning computation and the resolution of trilateration. Then, we present our proposals in terms of mobility management. Finally, we determine the requirements for a mobility management service.

A. The FBCM and RADAR-like based Hybrid Model

On map 2, the triangles are the access points. The dots are the calibration measurement points and the crosses are the testing points. The origin of the coordinates is the bottom left corner on the map. The singular points observed in Interlink Networks project [9] are due to the heterogeneity of the topology. That require we adapt the Friis equation proposed by Interlink Networks.

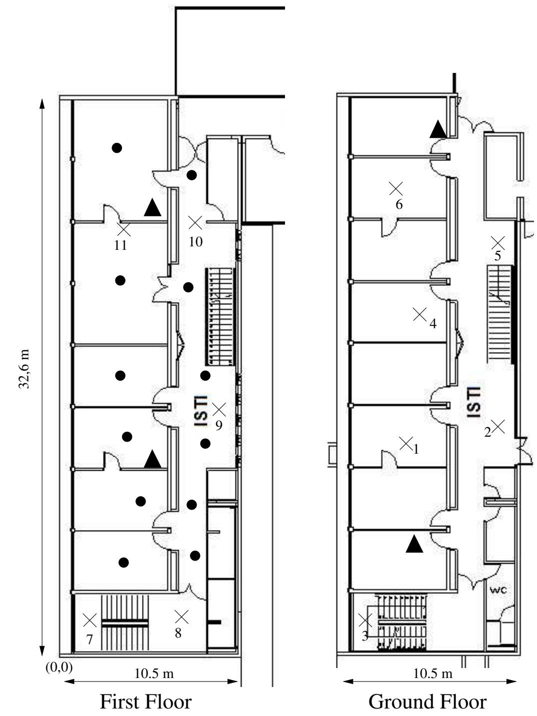

Fig. 2. Ground and first floor of the ISTI wing of Numerica.

We compute the distance according to a coefficient adapted to the topology. The coefficient replaces the square of the distance in the Friis equation. It is determined by manipulating the Friis equation to obtain an expression of the coefficient according to the distance and the signal strength. The coefficient i_{jk} is the coefficient at the point j for the access point k. This coefficient is determined as follows: $i_{jk} = \frac{-SS_{jk}-K}{10\log(d_k)}$ where SS_{jk} is the signal strength measured at the point j for the access point k, K is a constant value depending on hardware parameters and d_k is the real distance to the access point k. The distance is obtained by the following expression: $d = 10^{\frac{-SS-K}{10i}}$ with same variables as above.

The low accuracy observed while testing the FBCM is easy explained by the heterogeneous topology of our testbed. On the whole building, the Friis coefficients are not similar. Thus, by using the mean value of the Friis coefficients to determine the distance between the mobile's position and the access points, we commit a great error in the positioning process. To address this problem, the Friis coefficient used for the distance computation must be closer from the reality.

We make the assumption that the measurements for calibration of the FBCM will not greatly vary over time (for example due to people movements). It is true because the number of persons in the building remains approximately constant.

The FBCM is merged with a RADAR-like algorithm. Indeed, the accuracy of the FBCM is weak in a strongly heterogeneous environment, like our testbed. The goal of an hybrid model is to restrain the area of research to a sub-space where the Friis coefficients are close together and with the reality. It ensures that the distance computation will be realistic.

Data:
    DB: database of Scans
    M: list of measurements
    VSk: vector of k Scans
    H: set of $n-1$ vector of k Scans
    P_1, P_2: Points
    V: vector of Scans
    D: vector of distances

Begin
    Order DB Scans by $(ss_1 - m_1)^2 + ...$
        $... + (ss_n - m_n)^2$ ascending;
    Store k first Scans in VSk;
    $P_1 \leftarrow$ Viterbi(VSk, H);
    V $\leftarrow$ Neighbour Scans from P_1 in DB;
    D $\leftarrow$ Compute Distances for V;
    $P_2 \leftarrow$ Trilateration(P_1, V, D);
End

Fig. 3. Positioning Process Algorithm

In the RADAR-like algorithm (see figure 3), the scans in the database DB identify the reference points by the t-uple (x, y, z, ss_1 to ss_n, i_1 to i_n). The 3-uple (x, y, z) is the coordinates of the point. The values ss_1 to ss_n are the signal strength measured and i_1 to i_n are the corresponding Friis coefficients. The measurements list M is a vector (m_1 to m_n, ap_1 to ap_n) where the m_i are the signal strength measurements and the ap_i are the access points' identifiers. The couple (m_i, ap_i) is the signal strength measurement for the access point i.

In further explanations, we refer to the mobile terminal which needs to be located as the mobile.

On each positioning iteration, a set of signal strength measurements is established. It is either client-centric or infrastructure-centric.If client-centric, the procedure consists in scanning the 802.11 channels for access points beacons and sending the result to the positioning server. If infrastructure-centric, several steps are involved: first, the client broadcasts an UDP packet. It is composed of a positioning request code, the mobile's identifier, generally its mac address, and a timestamp. Each access points in range measures the signal strength of the UDP packet when receiving it. The access point's identifier (eg. its mac address) as well as the signal strength measured are concatenated to the UDP packet and sent to the measurement server. When the measurement server receives a packet with a mobile identifier and a timestamp never seen, it starts waiting others packets with the same mobile identifier and timestamp until a timeout expires. Then, all packets are grouped and sent to the positioning server.

Now comes the positioning process. The set of measurements in the final packet is compared to the reference points database. The K closest points in the signal strength space are selected. In a technical report [16], the authors of the RADAR system added a Viterbi-like algorithm to improve their accuracy. We use a Viterbi-like algorithm too, but the distances we use are real distances.

Before exposing the distance computation, we need to introduce the way we model the physical space. We think in terms of positioning site. A positioning site is a sub-space in which the positioning system is operational. Each positioning area is not connected with another one. Basically, a set of buildings covered by a WLAN can be a positioning site. Positioning sites are composed of buildings, themselves composed of floors. The floors contain reference points. This topology is stored in a relationnal database.

We determine the distances by setting a graph of the nearest neighbours. We represent it by a matrix giving for each point the distance towards its nearest neighbours. The other distances are set to $+\infty$. Then, each iteration extends the distances to the others points by adding the distance between a point and its neighbour to the distance between this neighbour and one of its own neighbours. Each iteration can only replace a distance by a smallest one. Thus, we obtain the matrix of minimal distances between every points. The distances are computed only for points in the same floor. It reduces the size of the set of distance matrices but prevents the positioning systems to track a mobile which changes its floor.

The distances of the matrix and the K points selected to run the Viterbi-like algorithm allow to choose one point. All reference points of its neighbourhood (defined by a distance parameter) are selected. For each point, the difference between the distance computed with its own coefficient and the real distance is computed. Then, the iterative trilateration algorithm presented in [11] gives a mobile's location in the neighbourhood of the point chosen. This model is named *FBCM and RADAR-like based Hybrid Model*.

In order to predict the future locations of a mobile, we use the Kth Markov Model described in [15]. We improve it by selecting a set of possible locations according to a transition probability threshold. It allows to prefetch or plan a handover on several new locations, thus improving the success probability of the model. We call this model the *Threshold-based Kth Markov Model* (TKMM).

B. Service requirements

The Mobility Service (MS) requires several conditions. First of all, one has to determine a protocol between the MS and the applications using it. In the same way, the system component which computes the distances between the mobile terminal and the wireless access points has to be dissociated. Then one needs an interface to communicate between this component and the MS.

The MS is conceivable as a middleware whose its tasks are limited, first to compute the trilateration, and second to manage the mobility (prediction and handover). Then the use of such a service becomes transparent for applications, and the MS running remains the same whatever the WiFi card to be considered. Thereafter, it is possible to extend the MS to other networks (bluetooth, UMTS, etc).

An MS does not have to neglect neither the auto-configuration nor the auto-learning. So, the configuration is made during the system installation. Later on, the auto-learning is effective when transition probabilities are updated. The transition probabilities can also be modified when considering a new arrival of mobility patterns.

C. GeoMoVie architecture

In figure 4, we detail the GeoMoVie architecture. It is a middleware which role is to locate mobile terminals. It is a multithreaded service. At the beginning, the database server and the main thread run on the server. On step 1, a mobile client sends a positioning request to the positioning server. Then, the main thread runs a new thread (step 2a) and sends a measurement request (step 2b) to either the access points (infrastructure-side measurements) or the mobile client itself (client-side measurements). On step 3, the wireless drivers of either the access points or the client sends back a measurement response, which contains a list of access points mac addresses and signal strength. On step 4, the client thread sends the SQL request to the database server. The server returns a list of points close to the measurement on step 5. Then, the client thread runs the Viterbi-like algorithm and determines the final location with the FBCM. On step 6, the location is send to the mobile. The location-aware services running on the mobile terminal can use the position received to continue their execution (mobility prediction, cache prefetching).

For now, we use GeoMoVie on the MoVie project and in GuiNuMo. GuiNuMo is a digital mobile guide for museums. Both applications need location informations to decide which multimedia content to stream to their clients.

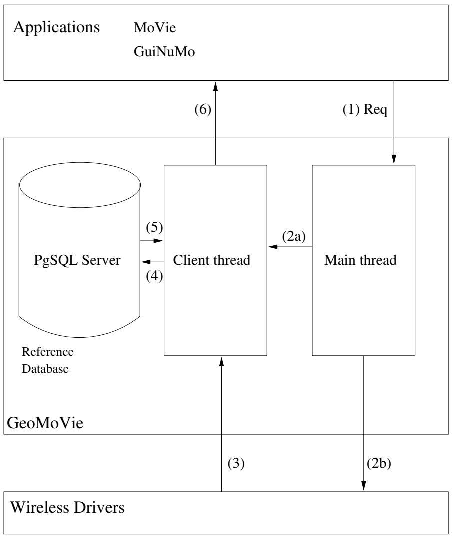

Fig. 4. GeoMoVie architecture.

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

in table I, we expose the results of the FBCM alone. One can see that the accuracy is not sufficient to provide location-based services. The improvement of the Hybrid Model is visible in table II.

Point index (fig. 2)	Interlink Net. error [9] (m)	SNAP-WPS error [8] (m)	Our model error (m)
1	37,43	n/a	**33,68**
2	44,06	n/a	**30,05**
3	**30,79**	n/a	43,48
4	36	n/a	**27,4**
5	37,17	n/a	**21,87**
6	27,9	n/a	**23,44**
7	15,7	20,44	**2,72**
8	20,08	19,62	**7,62**
9	24,08	23,89	**16,86**
10	48,87	**17.29**	26,32
11	30,6	**6,58**	25,06
12	36,95	48,86	**16,56**

TABLE I. Precision of the positioning models.

The measurements and the computations made by the positioning system allow us to analyze the accuracy of the related work ([8] et [9]) and the accuracy of the calibrated model. The accuracy measurement is simple.

True coordinates	Positioning	Error (meters)
(5.8,4.4)	(6,3.2)	1.2
(7.8,21.4)	(8.6,13.4)	1.3
(7.6,25.2)	(7.6,24.6)	0.7

TABLE II. Sample of the tests on the hybrid model.

The positioning system computes the mobile terminal location and the euclidian distance between the result and the true position which is provided with the signal strength measurements. That is why we used points with known coordinates. The coordinates are simply measured in the building.

The measurements are carried out on the ground floor and on the first floor in Numerica. The results obtained are given in table I.

The calibrated model accuracy is in most cases better than the accuracy of the models presented by Interlink Networks [9] and SNAP [8]. The points where the accuracy of the calibrated model is less than that of the others projects are located behind heavy obstacles (ie. a load-bearing wall or in a stairwell). These locations are not part of the calibration area, which explains the lack of precision.

The absolute error made by the calibrated model on the ground floor is easy to explain. Indeed, the calibration was carried out on the first floor. The use on the ground floor creates huge distance miscalculation because the topology is different and the access points are not relatively located at the same coordinates. But the calibrated model remains competitive with the other positioning models. A quick overview of the results raises a question. How can the errors made by the three models be explained ? The errors are great compared to the building size. We came to the conclusion of topology influence on the positioning system.

It is easy to explain this fact mathematically. Let us consider an access point and two points of the plan at an equal distance d of the access point. A point has a line of sight with the access point. The other point is hidden by a wall. On each point, the signal strength measured or theoretical is different. It shows that the relation between the distance and the signal strength is not bijective. Therefore, this relation does not admit a reciprocal relation. The reciprocal expressions of the Friis equation and its alternatives are approximate compared to the real conditions. Therefore, it can be concluded that, the more heterogenous the topology, the more inaccurate using a reciprocal expression based on the Friis equation is.

The hybrid model's accuracy is satisfying to provide location-aware services, such as mobility management. Its accuracy if far better than that of the related works models or the FBCM. However, the calibrated model has been tested on one floor only, because the computation of the distance is made on each floor independantly. Thus, the logical topology between floors and buildings is not considered yet.

B. The Threshold-based Kth Markov Model

We tested the TKMM with 2 context databases provided by the **Institut fur Pervasive Computing**[1]. One is entitled *benchmark* and the other is entitled *Nokia Context Data*. The results are shown in tables III and IV.

Probability threshold	1-past	2-past	3-past	4-past
1	0.07	0.2729	0.2705	0.2077
0.9	0.1546	0.3551	0.343	0.2754
0.8	0.2488	0.4589	0.4324	0.3527
0.7	0.2874	0.5459	0.4831	0.3792
0.6	0.3092	0.5676	0.4976	0.3816
0.5	0.4952	0.6618	0.5507	0.4082
0.4	0.587	0.6667	0.5556	0.4082
0.3	0.7029	0.7198	0.5749	0.4251
0.2	0.7657	0.756	0.5942	0.4251
0.1	0.8647	0.8213	0.6135	0.4324
0	0.9469	0.843	0.6256	0.4444

TABLE III. Mobility prediction accuracy, Nokia Context Data.

Probability threshold	1-past	2-past	3-past	4-past	10-past
1	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	0.46	0.553	0.132
0.4	0.4879	0.5279	0.5074	0.5967	0.1366
0.3	0.4879	0.619	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	0.9898	0.9665	0.921	0.8411	0.1961

TABLE IV. Mobility prediction accuracy, benchmark.

Tables III and IV expose the accuracy on, respectively, the Nokia Context Data and benchmark when using only one degree of Markov Model. We can draw several conclusions from these results. First, we see that the lower the probability threshold, the better the accuracy is. It is easy to explain, because the lowering of the threshold leads to an increasing of the number of neighbour states prepared for the handover, which implies a greater probability of success.

Second, we see that the 1-past accuracy is better than the other's ones below a given threshold. This is explained because 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, we see that 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 fact is not absolute truth. There may be some cases for whose the precision would go growing with the increase of the automaton's degree.

Fourth, for several thresholds, the accuracy remains the same and then suddenly increases. It is due to the existence of a transition very often taken compared to the other ones. Thus, the transition mostly taken 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 benchmark data set, which represents movements in an office building, it means that a user 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 IV, 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.

V. Conclusions

In this paper we presented a mobility management middleware. The middleware consists in two main parts: the positioning of the mobile terminals and the mobility prediction. These two methods was applied in the context of a Wifi network. In spite of the numerous techniques of outdoor positioning which exist, indoor positioning is rarely implemented and when it is, precision is rarely good. We proposed some elements of indoor positioning improvement. We achieve the positioning stage with the *FBCM and RADAR-like based Hybrid Model* which combines a reference points approach with a radio wave propagation model. We showed that its accuracy is better compared to related works models and our previous proposals. However, the positioning system is not complete yet. It must take into account the buildings' logical topology.

A kind of Kth Markov Model is tested. We have

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

combined it 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. However, much improvements can be made to this model.

We also described the architecture for a mobility management middleware and we exposed the advantages of this approach. We have also exposed the limits of this architecture as far as we went.

The applications of a middleware of management of mobility are numerous. Mobile digital guides, for example, are applications very dependent on the localization. The mobility prediction particularly applies to handover anticipation. The latter ensures continuity of the services with more reliability. In particular, our middleware is used in the MoVie project [2]. The GeoMoVie component manages mobility and allows the continuity of the streaming of multi-media flows.

VI. Future work

The mobility prediction needs improvements. First, the multimedia streaming and the handover 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, the increase of the *K-to-1 past* model's degree sometimes decreases the accuracy. We think about combining every Markov Model's degree to make a prediction. Thus, the accuracy should only grow with the increase of the model's degree. The improvements described above are short-term improvements.

The way we build mobility patterns can be improve. 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.

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

[3] 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.

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

[5] M.A. Youssef, A. Agrawala, A.U. Shankar, and S.H. Noh. A probabilistic clustering-based indoor location determination system. Technical report, Maryland Information and Network Dynamics Laboratory, March 2002.

[6] Z. Xiang, S. Song, J. Chen, H. Wang, J. Huang, and X. Gao. A wireless lan-based indoor positioning technology. *IBM Technical Journals*, 48(5), 2004.

[7] R. Battiti, A. Villani, and T. Le Nhat. Neural network models for intelligent networks: deriving the location from signal patterns, 2002.

[8] 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.

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

[10] L.V. Blake. *Radar Range-Performance Analysis*. Artech House Radar Library, December 1986.

[11] 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.

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

[13] 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.

[14] 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.

[15] 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*, volume V, pages X–Y. USENIX, USENIX, October 1999.

[16] P. Bahl, A. Balachandran, and V. Padmanabhan. Enhancements to the radar user location and tracking system. Technical report, 2000.