

Evaluating Filtering Strategies for Decentralized Handover Prediction in the Wireless Internet

Paolo Bellavista, Antonio Corradi, Carlo Giannelli
Dip. Elettronica, Informatica e Sistemistica - Università di Bologna
Viale Risorgimento, 2 - 40136 Bologna - ITALY
Phone: +39-051-2093001; Fax: +39-051-2093073
{pbellavista, acorradi, cgiannelli}@deis.unibo.it

Abstract

The rapid diffusion of heterogeneous forms of wireless connectivity is pushing the tremendous growth of the commercial interest in mobile services, i.e., distributed applications to portable wireless terminals that roam during service provisioning. In the case of both location-dependent mobile services and mobile services with session continuity requirements, there is a growing need for decentralized and lightweight solutions to predict cell handovers, in order to enable proactive service management operations that anticipate actual terminal reconnections at their newly visited cells. The paper discusses how to predict client handovers between IEEE 802.11 cells in a portable and completely decentralized way, only by exploiting RSSI monitoring and with no need of external global positioning systems. In particular, the paper focuses on proposing and comparing different filtering techniques for mitigating Received Signal Strength Indication abrupt fluctuations. Experimental results point out that i) filtering techniques can relevantly improve the efficiency and effectiveness of handover prediction, and ii) the choice of the most appropriate filtering solution to adopt should be made at provisioning time depending on specific service/system requirements, e.g., privileging minimum overhead vs. greater prediction proactivity.

1. Introduction

The increasing availability of mobile devices with wireless communication capabilities stimulates considering new service provisioning environments where accessibility to the traditional Internet is provided via Access Points (APs) working as bridges between fixed hosts and wireless devices [1]. The most notable example is the case of IEEE 802.11 APs that support

connectivity of Wi-Fi terminals to a wired local area network [2]. In the following, we will indicate these integrated networks with fixed Internet hosts, wireless terminals, and wireless APs in between, as the Wireless Internet (WI).

WI opens new challenging scenarios for mobile service provisioning. On the one hand, time-continuous services, such as audio/video streaming, relevantly suffer from the temporary disconnections that mobile clients experience at their handovers from old Wi-Fi access localities to new ones; in addition, they often require moving reached session states in newly visited localities. On the other hand, WI pushes towards the possibility to provide novel services whose contents depend on client location; location dependency complicates application design and implementation, and requires innovative support functions [3].

We claim that both mobile services with session continuity requirements and location-dependent ones can relevantly benefit from the adoption of lightweight and decentralized mechanisms capable of predicting wireless client handover between WI access localities. In particular, the idea is to provide support functions for i) *handover prediction*, to notify interested services that one of their clients is going to change AP in a given time interval, and ii) *mobility prediction*, to notify interested services that one client is going to leave its current access locality by specifying which is the most probable next AP. For instance, handover prediction could help continuous services to pre-fetch data chunks just before handovers, thus simplifying service interruption avoidance while preserving wireless client memory when handovers are not probable [4]. Mobility prediction can permit to perform location-dependent service management operations, e.g., resource re-bindings, in next WI localities before the actual connections of mobile clients [3].

Our work on handover and mobility prediction is

part of our original middleware for the support of WI service provisioning, which transparently mediates wireless client access to distributed applications via mobile proxies that dynamically adapt service results to client terminal properties, location, and runtime resource availability [2,5,6,7].

The paper specifically discusses how to predict client handover/mobility between IEEE 802.11 cells in a portable and completely decentralized way, with no need of external global positioning systems. The solution guideline is to only exploit the standard monitoring information about Received Signal Strength Indication (RSSI) for both the currently connected AP and all other APs in client visibility. In particular, the paper presents, analyses, and quantitatively compares different techniques (Grey Model, Fourier Transform, Discrete Kalman, and Particle) for filtering RSSI fluctuations due to signal noise, by pointing out how filtering can positively impact on handover/mobility prediction performance.

The paper also includes a thorough experimental validation of the proposed mechanisms in a wide-scale simulated environment, which can model nodes randomly roaming among IEEE 802.11 APs. Reported results show that the performance of our handover/mobility predictor notably improves when adopting RSSI filtering techniques. In addition, the choice of the most appropriate filtering mechanism should be performed at service provisioning dependently on specific application requirements, for instance in order to privilege either minimum overhead or greater advance time in the handover/mobility prediction.

2. Handover and Mobility Prediction

The goal of our handover/mobility prediction solution is to provide information about the probability a handover process is going to start and to which AP the involved mobile client is going to connect at. To precisely describe how our handover/mobility prediction mechanism performs, it is first necessary to exactly clarify how communication-level handover works. In fact, the IEEE 802.11 standard does not impose any specific handover strategy: that permits network equipment manufacturers to be free to implement their own handover strategies, as detailed in the following. The different communication-level handover strategies in the market motivate different variants of our handover/mobility prediction mechanism: therefore, the paper proposes and compares two implementations specifically designed for the two most relevant classes of possible handover strategies, i.e., Hard Proactive (HP) and Soft Proactive (SP).

2.1. Communication-level Handover: Reactive and Hard/Soft Proactive Strategies

Cell-based wireless communications can adopt diverse strategies for communication-level handover, which mainly differ in the event used to trigger the handover process. In particular, it is possible to identify two main handover classes: reactive and proactive. *Reactive handover* tends to delay handover as much as possible: handover starts only when wireless clients completely lose their current AP signal. These strategies are effective in minimizing the number of handovers, e.g., by avoiding to trigger a handover procedure when a client approaches a new wireless cell, without losing the origin signal, and immediately returns back to the origin AP. However, reactive handovers tend to be long because they include looking for new APs, choosing one, and asking for re-association only after having lost previous AP signal.

Proactive handover, instead, tends to trigger handover before the complete loss of origin cell signal, e.g., when the new cell RSSI overpasses the origin one. In general, these strategies are less effective in reducing the number of useless handovers, but are prompter by performing search operations for new APs before the handover procedure starts. By concentrating on proactive handover, a further classification is possible. On the one hand, HP strategies trigger a handover any time the RSSI of a visible AP is greater than the RSSI of the currently associated AP plus an Hysteresis Handover Threshold (HHT); HHT is introduced mainly to prevent heavy bouncing effects. On the other hand, SP strategies are “less proactive” in the sense that they trigger handover only if i) the HP condition applies (there is an AP with RSSI greater than current AP RSSI plus HHT), and ii) the current AP RSSI is lower than a Fixed Handover Threshold (FHT).

For instance, the handover strategies implemented by Cisco Aironet 350 and Orinoco Gold Wi-Fi cards follow, respectively, the HP and SP models. More in detail, Cisco Aironet 350 permits to configure its handover with the “Scan for a Better AP” option: if the current AP RSSI is lower than a settable threshold, the Wi-Fi card monitors RSSI data for all visible APs; for sufficiently high threshold values, the Cisco cards behave according to the HP model. Orinoco Gold cards exactly implements the SP strategy, without giving any possibility to configure the used thresholds.

2.2. Our Modular Solution for Handover/Mobility Prediction

We have developed a middleware for proactive session migration and location-dependent services in WI;

our middleware exploits handover/mobility prediction to proactively perform service management operations only in the WI access localities that are going to be visited by mobile clients [4]. Here, we specifically focus on the proposed prediction mechanisms and, in particular, on the impact of filtering techniques on prediction performance.

Our handover/mobility prediction solution is based on a pipelined architecture consisting of two modules. The first one (*Filter*) is in charge of filtering RSSI sequences to mitigate RSSI fluctuations due to signal noise. The second module (*Prob*) tries to estimate the probability a handover happens in the near future and which is the most probable next AP based on RSSI values provided at its input from Filter. The modular architecture of our predictor permits a completely separated implementation and deployment of Filter and Prob, thus simplifying the exploitation and experimentation of different filtering and handover/mobility prediction mechanisms, even dynamically composed at provision time by downloading the needed module code [3]. For instance, the experimental results in Section 4 will show the performance of our middleware when the Prob module is fed with either actual RSSI sequences (Filter is the identity function) or filtered RSSI values produced by 4 alternative filters, extensively described in the next section.

By delving into finer details, we have implemented two variants of the Prob module, one suitable for communication-level HP handovers and the other for SP ones. We have decided not to work on Prob prototypes for reactive strategies because of two reasons: first, handover prediction is less challenging in the case of reactive handovers than of proactive ones since the triggering of a reactive handover only depends on the RSSI data from one single AP; secondly, reactive communication-level handovers are of minor interest for services with session continuity requirements, given their longer time needed to complete handover [4].

The HP-variant of our Prob module is in the state: LowProb, if the filtered value for the current AP RSSI is greater than the filtered RSSI values for any visible AP plus a Hysteresis Prediction Threshold (HPT); HighProb, otherwise. The SP-variant of the Prob module can assume the following states: LowProb, if the filtered RSSI value for the current AP is greater than either a Fixed Prediction Threshold (FPT) or the filtered RSSI value for any visible AP plus HPT; HighProb, otherwise. Figure 1 represents filtered RSSI values for current and next APs, in proximity of a HP (left) and SP (right) handover. A wireless client, moving from the origin AP locality to the destination AP one, is first associated with the origin AP (white background), then with the destination AP (grey back-

ground).

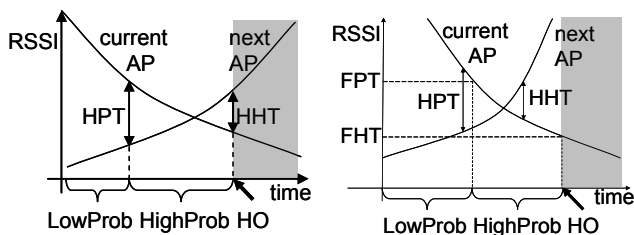


Figure 1 The different states of our two Prob variants: HP (left) and SP (right).

The performance of our prediction mechanisms can be quantitatively evaluated in terms of *hit rate*, *efficiency*, and *stability*. Informally speaking, hit rate estimates how many actual client handovers are correctly predicted, efficiency the capability to predict only client handovers that actually occur, and stability the ability to minimize Prob state changes. Our primary goal is to maximize hit rate to be able to proactively rearrange the new wireless access localities. In addition, as a secondary requirement, our prediction mechanism tries to maximize efficiency and stability. In fact, as better detailed in Section 4, both factors affect the overall system performance: depending on the type of provisioned service, handover/mobility predictions can produce service management operations of non-negligible overload, e.g., buffer migration and resource re-binding, and thus it is recommended to reduce useless predictions as much as possible.

3 RSSI Filtering

RSSI fluctuations due to signal noise significantly affect both stability and efficiency. For instance, in HP communication-level handover, when the RSSI value of the current AP is slightly greater than the sum of another AP RSSI plus HHT, even small RSSI fluctuations can produce several Prob changes, thus relevantly reducing stability. In addition, in those conditions RSSI over/under-estimation may trigger unnecessary predictions, thus lowering efficiency. The section presents 4 different filtering components we have implemented to mitigate RSSI fluctuations: Grey Model, Fourier Transform, Discrete Kalman, and Particle.

3.1 Grey Model

We have designed and implemented a first-order Grey Model filtering module that calculates filtered RSSI values on the basis of a finite series of RSSI values monitored in the recent past [8]. In particular, given one visible AP and the set of its actual RSSI

values measured at the client side $R_0 = \{r_0(1), \dots, r_0(n)\}$, where $r_0(i)$ is the RSSI value at the discrete time i , it is possible to calculate $R_I = \{r_I(1), \dots, r_I(n)\}$, where:

$$r_i(i) = \sum_{j=1}^i r_0(j) \quad (1)$$

Then, from the Grey Model discrete differential equation of the first order:

$$\frac{dr_i(i)}{di} + ar_i(i) = u \quad (2)$$

the wireless client can autonomously determine a and u , which are exploited to obtain the predicted RSSI value $pr(i)$ at discrete time i according to the Grey Model prediction function:

$$pr(i) = \left(r_i(1) - \frac{u}{a} \right) e^{-ai} + \frac{u}{a} \quad (3)$$

Let us observe that filtered RSSI depends on N , the number of actual RSSI values $r_0(i)$ employed in the Grey Model. In principle, greater N , more regular the RSSI filtered values, and slower the filtered RSSI sequence follows the possibly abrupt time evolution of actual RSSI. We have experimentally evaluated the Grey Model performance while varying N . The best trade-off between RSSI fluctuation mitigation and actual-to-filtered RSSI delay has demonstrated to be for $N=15$ in most common deployment scenarios. We used that value to obtain the reported experimental results (see Section 4).

3.2 Fourier Transform

Our Discrete Fourier Transform (DFT) filtering module extract from R_0 a Fourier coefficient set (A_i and B_i) representing the RSSI sequence in the frequency domain in a time window of duration (R_0 size)*(RSSI sampling period), with R_0 as defined in Section 3.1 [9]. The coefficient set is extracted with the usual Fourier equations:

$$A_0 = \frac{1}{N} \sum_{n=1}^N y(t_n) \quad B_0 = B_{N/2} = 0 \quad A_{N/2} = \frac{1}{N} \sum_{n=1}^N y(t_n) \cos(n\pi) \quad (4)$$

$$A_p = \frac{2}{N} \sum_{n=1}^N y(t_n) \cos\left(\frac{2\pi pn}{N}\right) \quad \text{where } p = 1 \dots N/2 - 1 \quad (5)$$

$$B_p = \frac{2}{N} \sum_{n=1}^N y(t_n) \sin\left(\frac{2\pi pn}{N}\right) \quad \text{where } p = 1 \dots N/2 - 1 \quad (6)$$

where $t_n = n\Delta t$, $\Delta t = \frac{T}{N}$, $\omega_p t = \frac{2\pi pn}{T}$, $N = R_0$ size

The Fourier coefficient set is the basis to define an Inverse Discrete Fourier Transform (IDFT) to regenerate the RSSI signal:

$$f(t_n) = \frac{1}{2} A_0 + \sum_{p=1}^M [A_p \cos(\omega_p t) + B_p \sin(\omega_p t)] \quad (7)$$

When IDFT exploits only a subset of the above se-

ries terms, the regenerated RSSI sequence do not exhibit its high frequency components and shows a more regular trend, i.e., IDFT behaves as a low pass filter. We have tested our Fourier Transform filter with several N values and different numbers of addends. We have found a good trade-off between fluctuation mitigation and actual-to-filtered delay with $N=4$ and $M=1$.

3.3 Discrete Kalman

Our Discrete Kalman filtering module tries to estimate RSSI values by representing the RSSI time evolution as a combination of signal noise (measurement noise) and maximum signal evolving (process noise) [10]. A linear stochastic equation models the RSSI evolution, with signal/process noise assumed to be independent of each other, white, and with normal probability distribution (standard deviation Q/R).

Our filter works by minimizing process noise (w) through a two phase algorithm: first, a predictor performs next RSSI estimation (equations 8 and 9); then, a corrector improves RSSI estimation by exploiting current RSSI measurement (equations 10, 11, and 12). Therefore, an iteration of our Discrete Kalman filtering module processes:

$$\hat{x}^-(k) = A\hat{x}^-(k-1) + w(k-1) \quad (8)$$

$$P^-(k) = AP^-(k-1)A^T + Q \quad (9)$$

$$K(k) = P^-(k)H^T (HP^-(k)H^T + R)^{-1} \quad (10)$$

$$\hat{x}(k) = \hat{x}^-(k) + K(k)(z(k) - H\hat{x}^-(k)) \quad (11)$$

$$P(k) = (1 - K(k)H)P^-(k) \quad (12)$$

where $P(k)$ is the covariance matrix of the state estimate error at step k , with initial value Q , and $K(k)$ is usually indicated as the Kalman Gain in the filtering literature.

In our scenario x and z are RSSI values, the state coincides with the output (A is a 1x1 identity matrix), and the estimation of the next state estimate is equal to the current state (H is a 1x1 identity matrix). After several tests, we have found a good trade-off between RSSI fluctuation mitigation and filtered-to-actual RSSI delay by setting $Q=1.6$ and $R=6$.

3.4 Particle

Like Discrete Kalman, our Particle filtering module tries to estimate RSSI by minimizing measurement and process noise, but without imposing a linear equation modeling and, more important in our deployment scenario, without imposing normal distribution for signal noise [11]. The main idea at its basis is to have an algorithm that computes, at each step, several possible filtered RSSI values for each measured RSSI; then, the filter associates each candidate value with a weight and

chooses the most promising values among them when a new measured RSSI is available; finally, it perturbs candidate values, according to the rules shown below, thus obtaining a new filtered RSSI (the average value of the most promising candidates).

To better and practically understand how Particle Filter works, let us show a rapid example of algorithm iteration with 10 particles, which represents 10 possible filtered RSSI values (Figure 2):

1. starting step - there are 10 possible filtered RSSI values (light circles), all with the same weight;
2. importance weight step - by exploiting state estimate probability (black curve) obtained from RSSI measurement, the algorithm assigns a weight at each filtered RSSI value (dark circles);
3. re-sampling step - heavy RSSI are spread in different RSSI values, all with the same weight (light circles), while light RSSI values are discarded;
4. sampling/prediction step - filtered RSSI are randomly perturbed (light circles).

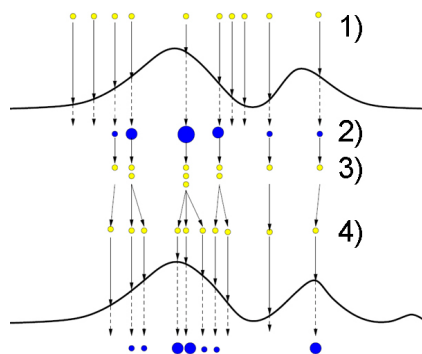


Figure 2 An example of Particle iteration.

The number of particles strongly influences the Particle filter performance; in general greater is the particle number, better the filtered RSSI follows the actual RSSI sequence. Differently from the previous 3 filters, Particle has a non-negligible computational load, which deeply depends on the particle number. For instance, by exploiting an Intel Pentium4 2.80GHz, 1024 MB of RAM machine, one iteration of our Particle filter takes about 132ms with 250 particles and 521ms with 500 particles; in a practical deployment scenario, where multiple APs are simultaneously visible from wireless clients, that time should be multiplied by the number of visible APs, and the computation occurs at any sampling interval. For the sake of completeness, Section 4 will report also Particle performance, when used with 250 particles and with the same Q and R as in Discrete Kalman; however, current client-side resource limitations discourage the exploitation of the Particle filtering module.

3.5 A Rapid Preliminary Comparison of RSSI Filtering Modules

Just to give a preliminary rough comparison of the behaviors of the proposed filters, Figure 3 reports 60 RSSI samples, either actually measured or filtered according to one of the 4 filtering modules. In particular, the figure points out how much filters are able to mitigate RSSI fluctuations and how fast filtered RSSI sequences follow the actual ones in the case of rapid RSSI evolving, e.g., samples 5 and 45.

Figure 3 is exemplar of the actual RSSI strong fluctuations (filter = Identity). Compared to actual RSSI, the output of the Grey Model has definitely less fluctuations, but tends to overestimate and to amplify RSSI growth in the case of rapid increasing, for instance when wireless clients are very close to their APs (see samples 10 and 50). Fourier and Kalman have similar behaviors: both tend to mitigate RSSI fluctuations quite well and accurately follow the actual RSSI sequence, without overestimating RSSI changes. Particle mitigates RSSI fluctuations very well, but sometimes introduces a non-negligible delay between actual and filtered values (see samples 8 and 47).

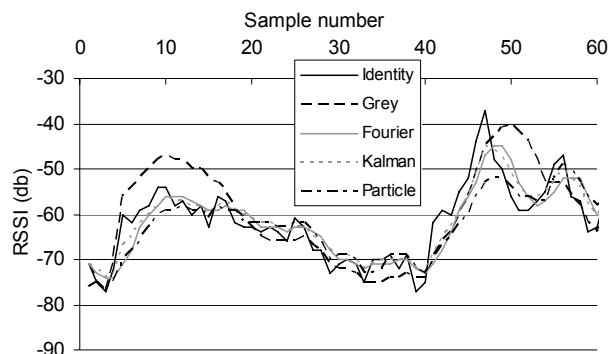


Figure 3 Actual and filtered RSSI.

4. Experimental Results

This section reports experimental results about the different hit rate, efficiency, and stability performance achieved when feeding our Prob module either with actual RSSI data or filtered RSSI (exploiting each time one of the 4 proposed filters). As already stated, the primary goal of RSSI filtering in our proposal is to mitigate RSSI fluctuations due to signal noise in order to primarily improve hit rate, with simultaneous acceptable values for efficiency and stability. However, better a filter mitigates RSSI fluctuations and longer is filtered-to-actual RSSI delay; in the following, for each filter we have adopted the parameter tuning described

in Section 3, which achieves a suitable trade-off between introduced delay and filtering performance.

In particular, in the case of handover prediction we have defined the following performance indicators:

- $hit\ rate = \left(\frac{HP_{pre}}{HP_{opt}} \right) * 100;$

where HP_{pre} is the total time elapsed in HighProb state in the t -long interval before handover and HP_{opt} is the time an optimal predictor should stay in High-Prob state (exactly t seconds before each handover). We have chosen $t=4s$ because such a time interval is largely sufficient to perform the needed service management operations in the WI localities going to be visited [3];

- $efficiency = \left(\frac{HP_{opt}}{HP_{tot}} \right) * 100;$

where HP_{tot} is the total time elapsed in HighProb state;

- $stability = \left(\frac{PC}{PC_{opt}} \right) * 100;$

where PC is the number of Prob state changes of our predictor and PC_{opt} the optimal number of Prob state changes.

On the contrary, in the case of mobility prediction, we are interested in evaluating:

- $hit\ rate = \left(\frac{CP}{NH} \right) * 100$

where CP is the number of correctly predicted handovers and NH is the total number of performed handovers;

- $efficiency = \left(\frac{CP}{NP} \right) * 100$

where NP is the total number of triggered predictions.

Note that we do not propose a stability indicator for mobility prediction since our proactive middleware for service management automatically inhibits further mobility predictions for a configurable time interval after a triggered prediction [4].

Obviously, efficiency and hit rate are strongly correlated: on the one hand, very good values for hit rate can be achieved at the expense of poor efficiency; on the other hand, it is possible to obtain very good efficiency by simply delaying as much as possible handover predictions, with the risk of missing handovers (too low hit rate). Moreover, it is necessary to maintain sufficiently high stability not to continuously perturb the service infrastructure with useless and expensive management operations.

We have measured the five indicators above in a challenging simulated environment, with a large number of Wi-Fi clients roaming among a large number of wireless APs (17 APs regularly deployed in a hexago-

nal cell topology); RSSI fluctuation has a 3dB standard deviation, $FPT=72dB$, $FHT=80dB$, $HPT=2dB$, $HHT=6dB$. Wireless clients follow movement trajectories according to the usual Random Waypoint model: speed is in the range [0.6m/s, 1.5m/s] and “thinking time” between 0s and 10s [12].

We have compared the behavior of the 4 proposed filtering modules within 6 scenarios (about 200 handovers performed in each one), differentiated for AP to AP distance (20m, 30m or 40m) and type of communication-level handover (HP or SP). For the sake of brevity, the paper reports the results for the most challenging deployment environment with greatest AP density (AP distance=20m); results about the other scenarios are rapidly commented in the following and extensively described at <http://lia.deis.unibo.it/Research/SOMA/MobilityPrediction>

Figures 4 and 5 reports average values of performance indicators for handover and mobility prediction, respectively. In general, the adoption of the proposed filtering techniques significantly improves stability, with relevant benefits from the point of view of system overhead due to useless Prob state changes. Efficiency also increases when exploiting filtered RSSI, except than in the case of Grey Model: in fact, Grey Model filtering tends to amplify the growth/decreasing trends of actual RSSI values, as observed in the previous section; that produces handover/mobility predictions with a large advance time but also with limited efficiency. Fourier, Kalman, and Particle filters, instead, tend to delay a little more Prob state changes, thus increasing efficiency. Let us observe that the hit rate performance indicator slightly decreases when adopting filtering techniques, except than in the Grey Model case. That effect is tied to what observed before: a slightly greater delay in handover prediction tends to weakly worsen hit rate, but with the relevant advantage of a significantly greater stability.

Similarly to handover prediction, the adoption of filtering techniques slightly lowers mobility prediction hit rate but, most important, increases its efficiency. Note that the reported efficiency is quite low, but it can be significantly improved with a better tuning of the Prob module configuration parameters e.g., HPT and FPT, as demonstrated in a previous work focused on Prob performance [13]. However, this paper specifically concentrates on pointing out the independent contribution on prediction performance of the 4 proposed filters and, for that reason, we have decided to exploit generic Prob settings here.

By taking a look at the whole set of reported performance results, it is clear that there is not a filtering technique always outperforming the others. Filtering exploitation has demonstrated to be crucial for increas-

ing stability; the decision on which filter to exploit depends on the specific goals of the provisioned mobile service. On the one hand, if it is of key relevance to achieve very high hit rates, the Grey Model is to be considered. On the other hand, if prediction costs must be as low as possible (high efficiency), Fourier and Kalman are good candidates.

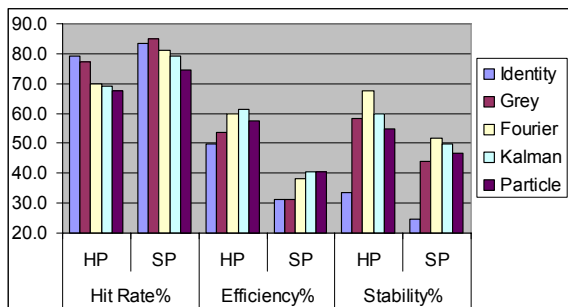


Figure 4 Handover prediction performance.

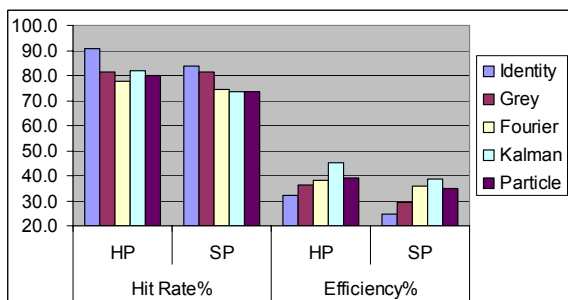


Figure 5 Mobility prediction performance.

Let us stress that, thanks to the modular architecture of our middleware, it is possible to choose the exploited filtering module at provisioning time, by adapting middleware behavior to the current service context. For instance, consider the case that our middleware exploits handover prediction to support the pre-fetching of data chunks before handover, in order to continuously provide service flows also during AP changes with temporary disconnections [4]. If the crucial point is to avoid temporary service interruptions, the best choice is Grey Model filtering; otherwise, if the priority is to minimize useless exploitation of client-side buffers, either Fourier or Kalman should be chosen. Finally, as already stated, Particle should be excluded from possible choices because of its high client-side computational load.

In addition to the proposed dense deployment scenario, we have evaluated our filtering modules in environments with greater AP-to-AP distance (30m/40m). The primary difference is a uniform increase in hit rate: when clients perform their handovers, APs are more distant and RSSI values exhibit a slower time

evolution, thus facilitating handover/mobility prediction. That confirms the suitability of presenting only the most challenging deployment scenario: the trend of performance indicators is the same, with no tight dependence on AP density.

5. Related Work

Several research activities have already addressed handover/mobility prediction in wireless environments [14], especially in cellular networks [15]. However, most of them propose solutions based on either usual movement patterns, or statistics of mobility habits, or the estimate of current position/speed. For instance, [16] predicts future location/speed by exploiting a dynamic Gauss-Markov model applied to current and historical movement data. [17] bases its trajectory prediction on paths followed in the recent past and on the spatial knowledge of the deployment environment, e.g., by considering admissible path databases. Note that exploiting these position prediction solutions as the basis for handover/mobility prediction requires coupling them with the full knowledge of maps with AP coverage areas. In addition, in open and extremely dynamic scenarios, with medium/short-range wireless connectivity, user mobility behaviors change very frequently and irregularly, thus reducing the applicability of mobility statistics.

Some other approaches in the literature have specifically focused on RSSI filtering. [18] exploits the Grey Model to decide when triggering handovers, by comparing filtered RSSI values with average/current RSSI ones. Differently from our proposal, [18] exploits RSSI filtering to improve communication-level handover, e.g., to reduce unnecessary bouncing, and not to perform handover/mobility prediction. Other researchers use Kalman and Particle filtering techniques to combine RSSI measurements with mobility models: the goal is to reduce noise impact on localization, and not to low-pass filter RSSI sequences to reduce signal fluctuation for handover/mobility prediction [19, 20]. The Fourier Transform is widely exploited to perform signal processing but, to the best of our knowledge, our proposal to use it in RSSI filtering is original.

In summary, if compared with the literature in the field, our middleware is definitely original in integrating a lightweight, portable, completely decentralized, and modular handover/mobility prediction solution, only based on RSSI data, and in exploiting several different filters with the main goal of increasing hit rate while ensuring good efficiency and stability.

6. Conclusions

The widespread utilization of wireless technologies pushes towards the adoption of handover/mobility prediction mechanisms to facilitate the support of services with session continuity requirements and/or dependent on client location. The proposed solution can not only perform prediction in a lightweight, scalable, and completely decentralized manner, but also adapt its behavior at provisioning time depending on service/system requirements, thus ensuring minimal intrusiveness and good stability. A relevant contribution to adaptivity is given by the possibility of dynamically choosing which RSSI filtering technique to adopt, thus influencing the expected performance of the prediction solution, e.g., by emphasizing either proactivity or efficiency.

The promising performance results already obtained are stimulating further related research activities. We are experimentally evaluating the performance of an improved version of our prediction mechanism that also exploits first-order RSSI derivative and chooses the filtering technique depending on a more refined RSSI evolution model; the goal is understanding whether more complicated RSSI models are justified by a relevant performance improvement. Moreover, we are evaluating the in-the-field behavior of our filtering techniques in the challenging application domain of WI multimedia streaming with session continuity requirements.

Acknowledgements

Work supported by the MIUR FIRB WEB-MINDS and the CNR Strategic IS-MANET Projects.

7. References

- [1] M.S. Corson, J.P. Macker, V.D. Park, "Mobile and Wireless Internet Services: Putting the Pieces Together", *IEEE Communications*, Vol. 39, No. 6, June 2001.
- [2] W. Stallings, "Wireless Communications and Networks", Pearson Education, Aug. 2001.
- [3] P. Bellavista, A. Corradi, R. Montanari, C. Stefanelli, "Context-aware Middleware for Resource Management in the Wireless Internet", *IEEE Trans. on Software Engineering*, Vol. 29, No. 12, Dec. 2003.
- [4] P. Bellavista, A. Corradi, C. Giannelli, "Adaptive Buffering based on Handoff Prediction for Wireless Internet Continuous Services", *Int. Conf. High Performance Computing and Communications (HPCC)*, Italy, Sept. 2005.
- [5] P. Ramanathan, K.M. Sivalingam, P. Agrawal, S. Kishore, "Dynamic Resource Allocation Schemes during Handoff for Mobile Multimedia Wireless Networks", *IEEE Journal on Selected Areas in Communications*, Vol. 17, No. 7, July 1999.
- [6] S. Saha, M. Jamtgaard, J. Villaseñor, "Bringing the Wireless Internet to Mobile Devices", *IEEE Computer*, Vol. 34, No. 6, June 2001.
- [7] K. Curran, G. Parr, "A Middleware Architecture for Streaming Media over IP Networks to Mobile Devices", *IEEE Int. Conf. Wireless Communications and Networking (WCNC)*, Mar. 2003.
- [8] J.L. Deng, "Introduction to Grey Theory", *The Journal of Grey System*, Vol. 1, No. 1, 1989.
- [9] P. Bloomfield, "Fourier Analysis of Time Series: An Introduction", John Wiley & Sons, 2nd edition, 2000.
- [10] G. Welch, G. Bishop, "An Introduction to the Kalman Filter", *SIGGRAPH 2001*, Aug. 2001.
- [11] R. van der Merwe, A. Doucet, J.F.G. de Freitas, E. Wan, "The Unscented Particle Filter", *Advances in Neural Information Processing Systems (NIPS13)*, Dec. 2000.
- [12] E. Hyttiä, J. Virtamo, "Random Waypoint Model in Cellular Networks", *Wireless Networks*, 2006.
- [13] P. Bellavista, A. Corradi, C. Giannelli, "Mobility Prediction for Mobile Agent-based Service Continuity in the Wireless Internet", *Int. Workshop on Mobility Aware Technologies and Applications (MATA)*, Brazil, Oct. 2004.
- [14] L. Song, D. Kotz, R. Jain, X. He, "Evaluating location predictors with extensive wi-fi mobility data", *23rd Conference of the IEEE Communications Society (INFOCOM)*, Hong Kong, China, Mar. 2004.
- [15] L. Perato, K. Al Agha, "Handover Prediction for Cellular Systems in a Multi-Services Context", *IEEE Int. Symposium on Wireless Personal Multimedia Communications (WPMC'02)*, Hawaii, USA, Oct. 2002.
- [16] B. Liang, Z.J. Haas, "Predictive Distance-Based Mobility Management for Multidimensional PCS Network", *IEEE/ACM Transactions on Networking*, Vol. 11, No. 5, Oct. 2003.
- [17] H.A. Karimi, X. Liu, "A Predictive Location Model for Location-based Services", *ACM Int. Workshop Advances in Geographic Information Systems (GIS)*, USA, Nov. 2003.
- [18] S.T. Sheu, C.C. Wu, "Using Grey Prediction Theory to Reduce Handoff Overhead in Cellular Communication Systems", *IEEE Int. Symp. Personal, Indoor and Mobile Radio Communications (PIMRC)*, UK, Sep. 2000.
- [19] J. Kwon, B. Dunder, P. Varaiya, "Hybrid Algorithm for Indoor Positioning Using Wireless LAN", *IEEE Vehicular Technology Conf.*, USA, Sep. 2004.
- [20] V. Seshadri, G.V. Záruba, M. Huber, "A Bayesian Sampling Approach to In-door Localization of Wireless Devices Using Received Signal Strength Indication", *IEEE Int. Conf. Pervasive Computing and Communications (PerCom)*, Hawaii, Mar. 2005.