Adaptive Buffering-based on Handoff Prediction for Wireless Internet Continuous Services

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.it1

Abstract. New challenging deployment scenarios are accommodating portable devices with limited and heterogeneous capabilities that roam among wireless access localities during service provisioning. That calls for novel middlewares to support different forms of mobility and connectivity in wired-wireless integrated networks, to provide runtime service personalization based on client characteristics, preferences, and location, and to maintain service continuity notwithstanding temporary disconnections due to handoff. The paper focuses on how to predict client horizontal handoff between IEEE 802.11 cells in a portable way, only by exploiting RSSI monitoring and with no need of external global positioning, and exploits mobility prediction to preserve audio/video streaming continuity. In particular, handoff prediction permits to dynamically and proactively adapt the size of client-side buffers to avoid streaming interruptions with minimum usage of portable device memory. Experimental results show that our prediction-based adaptive buffering outperforms traditional static solutions by significantly reducing the buffer size required for streaming continuity and by imposing a very limited overhead.

1 Introduction

The increasing availability of wireless Internet Access Points (APs) and the popularity of wireless-enabled portable devices stimulate the provisioning of distributed services to a wide variety of mobile terminals, with very heterogeneous and often limited resources. Even if device/network capabilities are always increasing, the development of wireless applications is going to remain a very challenging task. Their design should take into account several factors, from limited client-side memory to limited display size/resolution, from temporary loss of connectivity to frequent bandwidth fluctuations and high connectivity costs, from extreme client heterogeneity to local resource availability that may abruptly change in the case of client roaming. [1].

Let us consider the common deployment scenario where wireless solutions extend the accessibility to the traditional Internet via APs working as bridges between fixed hosts and wireless devices [2]. 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 [3]. 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 service provisioning must consider the specific characteristics of client devices, primarily their limits on local resources and their high heterogeneity. Limited processing power, memory, and file system make portable devices unsuitable for traditional services designed for fixed networks. These constraints call for both assisting wireless terminals in service access and downscaling service contents depending on terminal resource constraints. In addition, portable devices currently exhibit extreme heterogeneity of hardware capabilities, operating systems, installed software, and network technologies. This heterogeneity makes hard to provide all needed service versions with statically tailored contents and calls for on-the-fly adaptation. Client limits and heterogeneity are particularly crucial when providing *continuous services*, i.e., applications that distribute time-continuous flows of information to their requesting clients, such as audio and video streaming [4]. WI continuous services should address the very challenging issue of avoiding temporary flow interruptions when clients roam from one wireless locality to one another, also by considering client memory limitations, which do not allow traditional buffering solutions based on proactive client caching of large chunks of multimedia flows.

We claim the need of middleware solutions to support WI service provisioning to portable devices, by locally mediating their service access and by dynamically adapting results to client terminal properties, location, and runtime resource availability [3-6]. Middleware components should follow client roaming in different WI localities and assist them locally during their service sessions. Moreover, client memory limitations suggest deploying middleware components over the fixed network, where and when needed, while portable devices should only host thin clients. By following the above guidelines, we have developed an application-level middleware, based on Secure and Open Mobile Agent (SOMA) proxies, to support the distribution of location-dependent continuous services to wireless devices with limited on-board resources [7, 8]. The primary design idea is to dynamically deploy mobile proxies acting on behalf of wireless clients over fixed hosts in the network localities that currently offer client connectivity.

The paper focuses on an essential aspect of our middleware: how to avoid interruptions of continuous service provisioning when a client roams from one WI locality to one another (*wireless cell handoff*) at runtime. To achieve this goal, handoff prediction is crucial. On the one hand, it permits to migrate mobile proxies in advance to the wireless cells where mobile clients are going to reconnect, so to have enough time to proactively reorganize user sessions in newly visited localities, as detailed in a previous work [9]. On the other hand, service continuity requires maintaining client-side buffers of proper size with flow contents to play during the handoff process and to reconnect to mobile proxies in the new WI localities. Handoff prediction can enable the adaptive management of client buffers, by increasing buffer size (of the amount expectedly needed) only in anticipation of client handoffs, thus improving the efficiency of memory utilization, which is essential for portable devices. Let us observe that the proposed adaptive buffering, specifically developed for our mobile proxy-based middleware to avoid streaming interruptions, can help any class of WI applications that benefit from content pre-fetching in the client locality.

In particular, the paper presents how to exploit handoff prediction to optimize the

utilization of client-side pre-fetching buffers for streaming data. The primary guide-line is to provide handoff prediction-based adaptive management of client buffers with the twofold goal of minimizing buffer size and triggering pre-fetch operations only when needed. Our adaptive buffering exploits an original solution to predict handoffs between IEEE 802.11 cells, by using only client-side Received Signal Strength Indication (RSSI) from the wireless APs in visibility. Our prediction mecha-nisms originally adopt a first-order Grey Model (GM) to reduce RSSI fluctuations due to signal noise; they are portable, lightweight, and completely decentralized, and do not require any external global positioning system [8,10].

The paper also reports a thorough performance evaluation of our handoff predic-tion-based adaptive buffering in a wide-scale simulated environment, which can model nodes randomly roaming among IEEE 802.11 APs. In addition, we have col-lected in-the-field experimental results by deploying our system prototype over an actual set of Wi-Fi laptops. Experimental results show that our adaptive buffering outperforms not only traditional solutions based on statically pre-determined buffer size, but also adaptive dimensioning solutions based on non-GM-filtered RSSI val-ues, also when considering different implementations of communication-level hand-off by different Wi-Fi client card vendors. The proposed solution has shown to re-duce the buffer size needed to maintain streaming continuity and to impose a very limited overhead, by only exploiting local RSSI data already available at clients.

2 Related Work

Few researches have addressed position prediction in wireless networks, most of them by proposing solutions based on either the estimate of current position/speed or usual movement patterns. [11] predicts future location/speed by exploiting a dynamic Gauss-Markov model applied to the current and historical movement data. [12] bases its trajectory prediction on paths followed in the recent past and on the spatial knowl-edge of the deployment environment, e.g., by considering admissible path databases. Note that exploiting these position prediction solutions as the basis for handoff pre-diction requires coupling them with the knowledge of AP coverage area maps. In addition, in open and extremely dynamic scenarios, with medium/short-range wire-less connectivity, user mobility behaviors change very frequently and irregularly, thus making almost inapplicable handoff predictions based on past user mobility habits.

There are a first few approaches in the literature that have already investigated RSSI prediction. [13] predicts future RSSI values by using a retroactive adaptive filter to mitigate RSSI fluctuations; device handoff is commanded when the differ-ence between the current and the predicted RSSI values is greater than a threshold. [14] exploits GM to decide when to trigger the communication handoff by comparing RSSI predictions with average and current RSSI values. However, both [13] and [14] apply RSSI prediction to improve communication-level handoff, e.g., to reduce un-necessary bouncing, and not to predict the movements of wireless clients so to adap-tively manage their streaming buffers.

Adaptive buffer management for stream interruption avoidance is a well investi-gated area in traditional fixed distributed systems. [15] exploits predicted network

delay to adapt buffer size to the expected packet arrival time; its goal is to avoid stream interruptions due to packet jitter variations. [16] compares different algorithms for adaptive buffer management to minimize the effect of delay jitter: packet delay and/or packet peak recognition are exploited to dynamically adapt the buffer size. [17] predicts buffer occupation with a Proportional Integral Derivative predictor; the goal is to minimize packet loss due to buffer overflow. [18] is the only proposal dealing with wireless clients: it suggests a proxy-based infrastructure exploiting neural networks to predict client connection state; proxies execute on the wired network to manage the state/connections of their associated clients. Its primary goal is to preserve session state and not to maintain service continuity.

To the best of our knowledge, our middleware is definitely original in integrating a lightweight, portable, completely decentralized, and modular handoff prediction solution, only based on RSSI data, and in exploiting it to increase the efficiency of adaptive buffer management for continuous services.

3   Client-side Adaptive Buffering based on Handoff Prediction

Given the crucial role of efficiently managing client buffers to prevent streaming interruptions during client roaming, we propose an innovative buffer management solution that tends to save limited client memory by increasing buffer size only when a client handoff is going to occur. To this purpose, it is first necessary to exactly clarify how communication-level handoff works. In fact, the IEEE 802.11 standard does not impose any specific handoff strategy and communication hardware manufacturers are free to implement their own strategies, as detailed in the following. The different communication-level handoff strategies motivate different variants of our prediction-based adaptive buffering: for this reason, the paper proposes and compares two alternative buffer implementations specifically designed for the two most relevant classes of possible handoff strategies, i.e., Hard Proactive (HP) and Soft Proactive (SP).

3.1   Reactive and Hard/Soft Proactive Communication-level Handoff

Several communication-level handoff strategies are possible, which mainly differ in the event used to trigger handoff. We can identify two main categories: reactive and proactive. Reactive handoff strategies tend to delay handoff as much as possible: handoff starts only when wireless clients lose their current AP signal. Reactive strategies are effective in minimizing the number of handoffs, e.g., by avoiding to trigger a handoff process when a client approaches a new wireless cell without losing the origin signal and immediately returns back to the origin AP. However, reactive handoffs tend to be long because they include looking for new APs, choosing one of them, and asking for re-association. In addition, in reactive handoffs the RSSI of associated APs tends to be low in the last phases before de-association, thus producing time intervals before handoff with limited effective bandwidth. Proactive strategies, instead, tend to trigger handoff before the complete loss of the origin AP signal, e.g., when the new cell RSSI is greater than the origin one. These strategies are less effective in reducing

the number of useless handoffs but are usually prompter, by performing search operations for new APs before starting the actual handoff procedure and by limiting time intervals with low RSSI before handoffs.

By concentrating on proactive strategies, two primary models can be identified. On the one hand, HP strategies trigger a handoff any time the RSSI of a visible AP overcomes the one of currently associated AP of more than an Hysteresis Handoff Threshold (HHT); HHT is usually introduced to prevent heavy bouncing effects. On the other hand, SP strategies are "less proactive" in the sense that they trigger handoff 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 Handoff Threshold (FHT). For instance, the handoff 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 handoff strategy 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 of all visible APs, looking for a better AP; for sufficiently high threshold values, the Cisco cards behave according to the HP model. Orinoco Gold cards implements the SP strategy, applied to the Signal to Noise Ratio (SNR) in place of RSSI, without giving any possibility to configure the used thresholds.

3.2 HP- and SP-Handoff Predictors

The goal of our handoff prediction-based buffer management is to have client buffers of the maximum size and full exactly when actual re-associations to destination APs occur. Wrong handoff predictions produce incorrect dimensioning of client-side buffers; correct but late handoff predictions cause correctly-sized buffers that are not fulfilled with the needed pre-fetched streaming data.

Just to give a rough idea of the magnitude order of the advance time needed in handoff prediction, let us briefly consider the example of a client receiving a multimedia stream played at 1000Kbps constant bit-rate and a handoff procedure taking 1.5s to complete. That time interval includes the time for communication-level handoff and the time to locally reconnect to the migrated companion proxy, and largely overestimates the actual time measured in [8]. In this case, the client-side buffer size must be at least 187.5KB. If the client available bandwidth is 1500Kbps on average, the buffer fills with a speed of 500Kbps on average, by becoming full (from an empty state) in about 3s. Therefore, in the worst case, our handoff prediction should be capable of anticipating the actual handoff of 3s, to trigger buffer pre-fetching in time.

Our buffer management solution exploits a predictor structured in three pipelined modules. The first module (*Filter*) is in charge of filtering RSSI sequences to mitigate RSSI fluctuations due to signal noise. The second module tries to estimate handoff probability in the near future (*Prob*) based on RSSI values provided at its input. The last module (*Dim*) determines the correct buffer size to enforce, depending on handoff probability. More formally, our predictor consists of three modules, each of them implementing a function with the following domain and co-domain of definition:

- Filter : RSSI → FilteredRSSI

- Prob : FilteredRSSI $\rightarrow$ HandoverProbability

- Dim : HandoverProbability $\rightarrow$ BufferSize

The modular architecture of our predictor permits a completely separated implementation and deployment of Filter, Prob, and Dim modules, thus simplifying the exploitation and experimentation of different filtering, handoff prediction, and buffer size management mechanisms, even dynamically composed at provision time by downloading the needed module code [19]. For instance, the experimental results in Section 4 will show the performance of our middleware when feeding the Prob module with either actual RSSI sequences (Filter is the identity function) or GM-filtered ones (by pointing out the improvement due to the only GM-based RSSI filtering).

By delving into finer details about the already implemented predictor modules available in our middleware, Prob can assume three different states:

- LowProb, if handoff is considered highly improbable in the near future;
- HighProb, if handoff is considered almost certain in the near future;
- MedProb, otherwise.

Dim exploits the state delivered by Prob to dynamically modify the size of associated client-side buffers: in the current implementation, when in the HighProb state, Dim sets the buffer size at the maximum for that multimedia flow (flow bit-rate * 1.5s); when in LowProb, Dim sets the size at the minimum (maximum/10); and when in MedProb, it sets the size at (maximum+minimum)/2. We are currently evaluating if more complex processing functions for Prob and Dim modules (e.g., with finer granularity for the discrete states of handoff probability and buffer size) could improve our middleware performance; first results encourage to exploit simple and lightweight module functions, which can achieve the needed performance results with a limited computational overhead (see Section 4).

The Prob module runs at the client side, is completely decentralized, and only exploits local monitoring information about the RSSI of all IEEE 802.11 APs in current visibility. The awareness of RSSI monitoring data is achieved in a completely portable way over heterogeneous platforms, as detailed in Section 3.4. In particular, we have implemented two variants of our Prob module, one suitable for communication-level HP handoffs and the other for SP ones. We have decided not to develop Prob modules for reactive strategies because reactive handoffs are inherently unsuitable for continuous service provisioning environments, given their longer time needed to complete handoff. In addition, handoff prediction is less challenging in the case of reactive communication-level handoffs than when dealing with proactive ones: the triggering of a reactive handoff only depends on one AP RSSI data.

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 Superior Threshold (HST);
- HighProb, if the filtered value for the current AP RSSI is lower than at least one filtered RSSI value for one visible AP plus a Hysteresis Inferior Threshold (HIT);
- MedProb, otherwise.

Figure 1 represents filtered RSSI values for current and next APs, in proximity of a HP handoff. 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 des-

tination AP (grey background). The HP Prob module state changes from LowProb to MedProb and finally to HighProb. When the actual RSSI of destination AP overcomes the actual RSSI of origin AP plus HHT, the handoff is triggered (for the sake of simplicity, the figure considers filtered RSSI values equal to actual ones in HO).
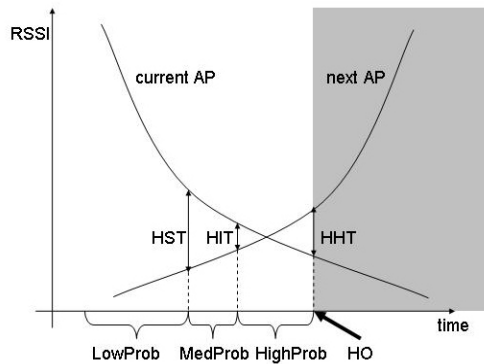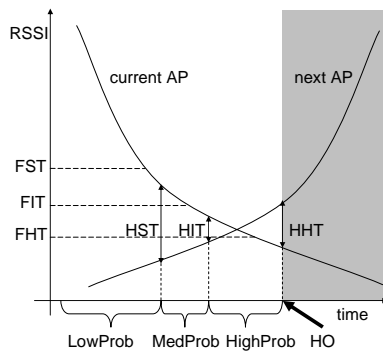

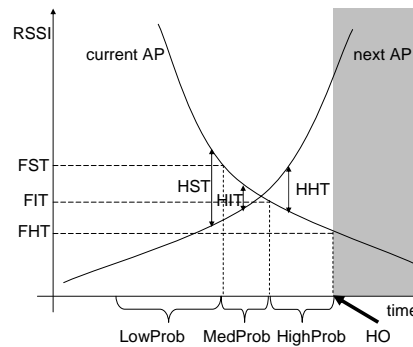**Figure 1.** The different states of the Prob HP-variant.

The SP-variant of the Prob module can assume the following states:
- LowProb, if the filtered RSSI value for current AP is greater than either a Fixed Superior Threshold (FST) or the filtered RSSI value for any visible AP plus HST;
- HighProb, if the filtered RSSI value for current AP is lower than a Fixed Inferior Threshold (FIT) and than at least one filtered RSSI of a visible AP plus HIT;
- MedProb, otherwise.

Similarly to Figure 1, Figures 2a and 2b represent filtered RSSI values for the origin and destination APs in proximity of an SP handoff. Figure 2a depicts a case where filtered RSSI values for current and next APs change relatively slow: in this case, changes in Prob results and actual handoff are triggered by the overcoming of hysteresis thresholds. In Figure 2b, instead, filtered RSSI values rapidly evolve and it is the passing of fixed thresholds that triggers Prob state variations and handoff.

Figure 2a. The different states of the Prob SP-variant for slow RSSI evolution.

Figure 2b. The different states of the Prob SP-variant for relatively fast RSSI evolution.

Both HP and SP variants of the Prob module take advantage of our GM-based Filter module. Filter exploits a very simple and lightweight first-order GM [10] to obtain filtered RSSI values on the basis of RSSI values monitored in the recent past, as described in the following section. Let us stress again that our modular handoff predictor is completely local and decentralized: each wireless client hosts its handoff predictor, whose Prob results only depend on either monitored or filtered RSSI values for all APs in visibility, with no need of interacting with any other client or SOMA-based middleware component running in the wired infrastructure.

3.3 Grey Model-based RSSI Prediction

Given one visible AP and the set of its actual RSSI values measured at the client side $R0 = \{r0(1), …, r0(n)\}$, where $r0(i)$ is the RSSI value at the discrete time i, it is possible to calculate $R1 = \{r1(1), …, r1(n)\}$, where:

$$r_1(i) = \sum_{j=1}^{i} r_0(j) \tag{1}$$

Then, from the GM(1,1) discrete differential equation of the first order [8]:

$$\frac{dr_1(i)}{di} + ar_1(i) = u \tag{2}$$

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 GM(1,1) prediction function [10]:

$$pr(i) = \left(r_1(1) - \frac{u}{a} \right) e^{-ak} + \frac{u}{a} \tag{3}$$

Let us observe that the average accuracy of the RSSI prediction may also depend on the number of actual RSSI values $r_0(i)$ employed by the adopted GM(1,1). In principle, longer the finite input series R_0, more regular the RSSI predicted values, and slower the speed with which the GM(1,1) prediction anticipates the actual RSSI sequence in the case of abrupt evolution [10]. We have evaluated the performance of our predictors, as extensively presented in Section 4, also while varying the number n of values in R_0. We did not experience any significant improvement in the predictor performance, on average, by using n values greater than 15. For this reason, all the experimental results reported in the following will refer to the usage of R_0 sets with 15 past RSSI values.

4   Experimental Results

To quantitatively evaluate the effectiveness of the proposed modular handoff predictor and of its application in our adaptive buffer management infrastructure, we have identified some performance indicators and measured them both in a simulated environment, with a large number of Wi-Fi clients roaming among a large number of wireless AP localities, and in our campus deployment scenario, where four laptops

move among the different coverage areas of six APs. Two laptops are Linux-based, while the other two host Microsoft Windows.NET; they alternatively exploit Cisco Aironet 350 (HP handoff) and Orinoco Gold (SP handoff) IEEE802.11 cards. In addition, we have compared the performance of our HP/SP Prob modules when connected to either our GM(1,1) Filter function or an identity Filter function that provides output values identical to its input: the goal was of understanding the isolated contribution of the GM(1,1) Filter function to the overall performance of our adaptive buffer management infrastructure.

In particular, we have considered the following performance indicators:

- *Average Buffer Size (ABS)* $= \dfrac{1}{T}\displaystyle\int_{0}^{T} BS(t)dt$,

where *BS(t)* is the time-varying buffer size. In other words, ABS is the time-weighted average of the buffer size;

- *Average Buffer Duration (ABD)* $= \dfrac{1}{T}\displaystyle\int_{0}^{T} BD(t)dt$

where *BD(t)* is the time-varying validity of a chosen buffer size. In other words, *ABD* is the average time interval between two successive operations of buffer re-sizing;

- *Successful Handoff (SH%)* $= \left(1 - \dfrac{DH}{NH}\right) * 100$

where *DH* is the number of actual client handoffs and *NH* is the number of handoffs predicted by the proposed HP/SP predictors.

In general, the goal of an optimal buffer management solution is to contemporarily achieve minimum values for *ABS* and sufficiently large *ABD* values, with maximum $SH_{\%}$. Obviously, *ABS* and $SH_{\%}$ are strongly correlated: on the one hand, very good values for $SH_{\%}$ can be easily achieved with large *ABS* values; on the other hand, it is possible to obtain very low *ABS* values by simply delaying as much as possible the buffer size enlargement, but with the risk of streaming interruptions (too low $SH_{\%}$ value). Moreover, it is necessary to maintain sufficiently large *ABD* values not to continuously perform useless and expensive buffer re-size operations.

We have measured the three indicators above in a challenging simulated environment where 17 APs are regularly placed in a 62m x 84m area and RSSI fluctuation has a 3dB standard deviation. Wireless clients follow trajectories with a randomly variable speed and with a randomly variable direction (with a Gaussian component for the standard deviation of $\Pi/6$). The speed is between 0.6m/s and 1.5m/s to mimic the behavior of walking mobile users; FST = 66dB; FIT = 70dB; FHT = 80dB; HST = 10dB; HIT = 6dB; HHT = 6dB. On the average, each wireless client has the visibility of ten APs at the same time, which represents a worst case scenario significantly more complex than the actually deployed Wi-Fi networks (where no more than five APs are usually visible at any time and from any client position).

Table 1 reports the average results for the three performance indicators over a large set of simulations, each one with about 500 handoffs. For the video streaming exploited in all the experiments, the buffer size required to avoid interruptions in the case of static fixed dimensioning is 200KB. The most important result is that any proposed Prob module, when provided with either GM-filtered RSSI values or actual

RSSI values, significantly reduces *ABS* (between 27.5% and 33.5%), thus relevantly improving the client memory utilization. In addition, Prob modules fed with GM-filtered RSSI values largely outperform the cases with actual RSSI values, especially with regard to the *ABD* performance indicator. In fact, even if *ABS* has demonstrated to maintain good values in all cases, directly monitored non-filtered RSSI (with its more abrupt fluctuations) tends to trigger a higher number of useless handoff predictions and, consequently, more useless modifications in the enforced buffer size.

Table 1 Performance indicators for HP and SP predictors. In the case of static fixed buffer: ABS=200KB, SH%=100, and ABD=∞.

Handover Strategy	Filter Function	ABS (KB)	SH%	ABD (s)
HP	Identity	140	92.1	2.80
	GM(1,1)	133	92.8	5.20
SP	Identity	145	91.6	2.79
	GM(1,1)	138	97.5	5.66

Figure 3 points out the correlation between GM-filtered RSSI and buffer size. In particular, it depicts the time evolution of buffer size (dotted line) depending on both GM-filtered RSSI of the currently associated AP (grey line) and the greatest RRSI among the non-associated visible APs (black line). Let us stress that when the currently associated AP RSSI is significantly greater than RSSI from other APs, our buffer management infrastructure maintains buffer size at its minimum (20KB in the example); when the RSSI of the currently associated AP, instead, is similar to the RSSI of another AP, buffer size increases at its maximum (200KB); otherwise, our infrastructure works to manage a medium-sized buffer (110KB).
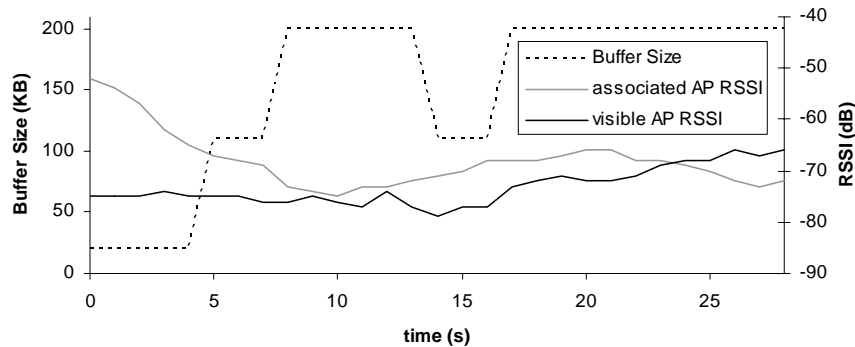
Figure 3. Buffer size variations depending on time evolution of GM-filtered RSSI values for the currently associated AP and for another AP in visibility.

In addition to simulations, we have evaluated the performance of HP/SP Prob modules also by using a service prototype, built on top of our middleware, and by moving four client laptops among the campus WI localities during streaming provisioning. Even if the number of considered in-the-field handoffs is largely lower than the simu-

lated one (thus, less relevant from the statistical point of view), in-the-field perform-ance results confirm the simulation-based ones. In particular, the prototype-based *ABS*, *SH$_\%$*, and *ABD* results have demonstrated to be better than simulation-based ones, on the average, also due to the lower number of considered APs, and the conse-quently simpler handoff prediction. However, we have experienced a significant degradation of prototype-based performance indicators in the case of extreme RSSI fluctuations, e.g., when a client follows a trajectory in strict proximity of relevant obstacles, such as the reinforced concrete walls of our campus buildings.

The code of the handoff prediction prototype, additional details about prototype implementation, and further simulation/prototype-based experimental results are available at `http://lia.deis.unibo.it/Research/SOMA/SmartBuffer/`

5   Conclusions and On-going Research

The exploitation of mobile middleware proxies that work over the fixed network on behalf of their resource-constrained clients is showing its suitability and effectiveness in the WI, especially when associated with handoff prediction. In particular, handoff prediction can help in realizing novel adaptive buffering solutions that optimize buffer size and pre-fetching. Our work of design, implementation, and evaluation of different buffering solutions has shown that our dynamic and simple GM-based pro-posal outperforms static buffering strategies, by preserving service continuity with limited requirements on client memory capabilities.

The promising performance results obtained are stimulating further related re-search activities. We are experimenting other handoff prediction techniques based on either higher-level GM models or the GM application to Ekahau-provided estimates of client positions (not directly to RSSI data) [20]. The goal is to evaluate whether a greater complexity of the Prob module of our predictor can significantly improve prediction quality, thus justifying the replacement of the currently adopted GM(1,1), which is extremely simple and lightweight.

Acknowledgements

References

1. Bacon, J., Bates, J., Halls, D.,: Location-oriented Multimedia. IEEE Personal Communica-tions, Vol. 4, No. 5, Oct. 1997.
2. Corson, M.S., Macker, J.P., Park, V.D.,: Mobile and Wireless Internet Services: Putting the Pieces Together. IEEE Communications, Vol. 39, No. 6, June 2001.
3. Stallings, W.: Wireless Communications and Networks. Pearson Education, Aug. 2001.

4. Ramanathan, P., Sivalingam, K.M., Agrawal, P., Kishore, S.: Dynamic Resource Allocation Schemes during Handoff for Mobile Multimedia Wireless Networks. IEEE Journal on Selected Areas in Communications, Vol. 17, No. 7, July 1999.
5. Saha, S., Jamtgaard, M., Villasenor, J.: Bringing the Wireless Internet to Mobile Devices. IEEE Computer, Vol. 34, No. 6, June 2001.
6. Curran, K., Parr, G.: A Middleware Architecture for Streaming Media over IP Networks to Mobile Devices. IEEE Int. Conf. Wireless Communications and Networking (WCNC), Mar. 2003.
7. Bellavista, P., Corradi, A., Stefanelli, C.: Application-level QoS Control and Adaptation for Video on Demand. IEEE Internet Computing, Vol. 7, No. 6, Nov.-Dec. 2003.
8. Bellavista, P., Corradi, A.: A QoS Management Middleware based on Mobility Prediction for Multimedia Service Continuity in the Wireless Internet. IEEE Int. Symp. on Computers and Communications (ISCC), July 2004.
9. Bellavista, P., Corradi, A., Giannelli, C.: Mobile Proxies for Proactive Buffering in Wireless Internet Multimedia Streaming. Int. Workshop on Services and Infrastructure for the Ubiquitous and Mobile Internet (SIUMI), June 2005.
10. Deng, J.L.: Introduction to Grey Theory. The Journal of Grey System, Vol. 1, No. 1, 1989.
11. Liang, B., Haas, Z.J.: Predictive Distance-Based Mobility Management for Multidimensional PCS Network. IEEE/ACM Transactions on Networking, Vol. 11, No. 5, Oct. 2003.
12. Karimi, H.A., Liu, X.: A Predictive Location Model for Location-based Services. ACM Int. Workshop Advances in Geographic Information Systems (GIS), Nov. 2003.
13. Kapoor, V., Edwards, G., Sankar, R.: Handoff Criteria for Personal Communication Networks. IEEE Int. Conf. Communications (ICC), May 1994.
14. Sheu, S.T., Wu, C.C.: Using Grey Prediction Theory to Reduce Handoff Overhead in Cellular Communication Systems. IEEE Int. Symp. Personal, Indoor and Mobile Radio Communications (PIMRC), Sep. 2000.
15. DeLeon, P., Sreenan, C.J.: An Adaptive Predictor For Media Playout Buffering. IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP), Mar. 1999.
16. Narbutt, M., Murphy, L.: Adaptive Playout Buffering for Audio/Video Transmission over the Internet. IEE UK Teletraffic Symposium, 2001.
17. Ip, M.T.W., Lin, W.W.K., Wong, A.K.Y., Dillon, T.S., Wang, D.: An Adaptive Buffer Management Algorithm for Enhancing Dependability and Performance in Mobile-Object-Based Real-time Computing. Int. Symp. Object-Oriented Real-Time Distributed Computting (ISORC), May 2001.
18. Kumar, B.P.V., Venkataram, P.: A Neural Network–based Connectivity Management for Mobile Computing Environment. Int. Journal of Wireless Information Networks, Apr. 2003.
19. Bellavista, P., Corradi, A., Montanari, R., Stefanelli, C.: Context-aware Middleware for Resource Management in the Wireless Internet. IEEE Trans. on Software Engineering, Vol. 29, No. 12, Dec. 2003.
20. Ekahau, Inc. – The Ekahau Positioning Engine v2.1. http://www.ekahau.com/products/positioningengine/