ESW4: Enhanced Scheme for WWW computing in Wireless communication environments

Stathes Hadjiefthymiades
Communication Networks Laboratory,
University of Athens, Department of Informatics
Panepistimioupolis, Athens 15784, Greece
Tel: (+301) 7275334
shadj@di.uoa.gr

Lazaros Merakos
Communication Networks Laboratory,
University of Athens, Department of Informatics
Panepistimioupolis, Athens 15784, Greece
Tel: (+301) 7275323
merakos@di.uoa.gr

ABSTRACT

Mobile computing is considered of major importance to the computing industry for the forthcoming years due to the progress in the wireless communications domain. In this paper, we present a proxy-based architecture, called ESW4, which manages to accelerate Web browsing in wireless CPNs. Proxy caches, maintained in base stations, are constantly relocated to accompany the roaming user. We discuss a cache management scheme involving the relocation of full caches to the most candidate cells but also percentages of the cache to less likely neighbors. Relocation is performed according to the output of a movement prediction algorithm based on a learning automaton. The simulation of ESW4 shows substantial benefits for the end user.

1. INTRODUCTION

Currently, the WWW [8] is viewed as a very promising technology and used vastly for the deployment of applications in the Internet and corporate intranets. This client/server information system, conceived in the early 90's, owns its great success in the standardization of the communication between browsers and WWW servers. The three open standards that are primarily involved in such communication are: the HyperText Transfer Protocol (HTTP), the HyperText Markup Language (HTML), and the Universal Resource Identifiers (URI) addressing scheme. Emerging standards like the XML [30] are also believed to give an even more impressive boost to the WWW.

Apart from the developments in the software area, during the 90's, we also witnessed tremendous advances in the area of wireless personal communications. The European cellular system GSM (Global System for Mobile Communications) [37] has received an unprecedented acceptance and spread rapidly over the globe. Office environments and small industrial installations have also benefited from the introduction of the Digital European Cordless Telecommunications standard (DECT) [18]. More sophisticated applications, enriched with multimedia capabilities (e.g., voice, VoD), have been made feasible with the HIPERLAN (High Performance Radio Local Area Network) standard [24]. A number of ATM based wireless LAN prototypes have been recently developed and discussed in the wireless networking literature [36], [17]. The near future is also very promising: the introduction of the fully fledged Universal Mobile Telecommunications System (UMTS) is planned for the period 2002 - 2005 [40]. This emerging standard will provide users with data rates up to 2 Mbps, circuit- and packet-switched service, and world wide coverage (satellite, macro-, micro-, or pico-cell environments). UMTS, combined with technologies like WAP (Wireless Application Protocol) [43] and VHE (Virtual Home Environment) [19], is an ideal platform for mobile multimedia.

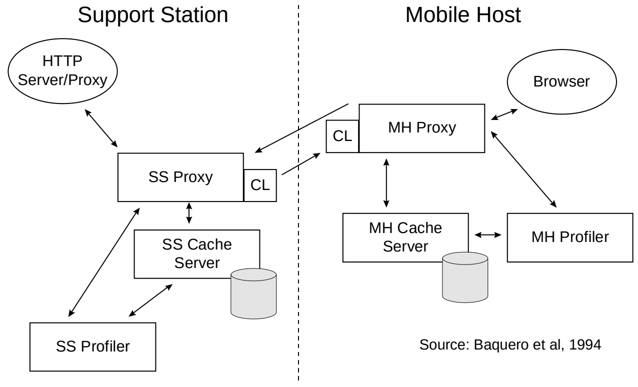
The growth of wireless telecommunications has stimulated the interest for the so-called anywhere - anytime computing. This type of computing, also known as "nomadic computing" [28], aims to provide users with access to popular desktop applications, applications specially suited for mobile users, and basic communication services. The emergence of nomadic computing was also facilitated by the rapid proliferation of portable computing equipment (portable PCs, portable digital assistants).

During the past years, a number of efforts were made to consolidate the WWW with wireless networking architectures (such integration is also referred to as W4 - World Wide Web for Wireless). Notable examples of such efforts are the MobiScape project [6] and the IBM Web Express platform [26], [20], which are used as a basis for our work[1]. Both systems capitalize on the proxy interface that most modern navigation tools support. The architecture developed in MobiScape is graphically shown in Figure 1.

The Mobile Host (MH) uses the Support Station (SS) as a gateway to the WWW. A caching mechanism is provided in both the MH and the SS in order to minimize the periods of connection between the two. The cache in the MH facilitates disconnected operation. The cache in the SS reduces the wait periods

[1] A more complete survey of such systems can be found in [23].

experienced when fetching remote documents. Thus, the HTTP data flow between the browser (MH) and the HTTP server is intercepted twice. Both the MH and SS cache servers operate as proxies [35]. The data flow between the SS proxy and the MH proxy is compressed (CL: Compression Layer). The profilers (incorporated in both proxies) consult user-defined scripts and trigger the pre-fetching of a series of documents that are very likely to be requested during the user's session. Novel ideas for a similar architecture (also termed "intercepting technology") are also proposed in the Web Express work. Furthermore, the Web Express platform addresses the requirements of transaction processing in Web applications through the adoption of "differencing" techniques.



Source: Baquero et al, 1994

Figure 1. Mobiscape architecture

The MobiScape/Web Express work, which is nicely harmonized with existing WWW software (i.e., does not require significant changes in browsers or servers) makes no provisions for the roaming of the user. If the wireless infrastructure provides for handovers between base stations (i.e., cellular environment), the SS cache has to be reconstructed each time the mobile terminal crosses the boundaries of a cell and establishes communication with a new base station.

In [22], we suggested an extension of the MobiScape/Web Express work. Specifically, we proposed the constant relocation of the SS cache so that it follows the movement of the mobile station. Cache relocation is performed prior to the realization of handovers according to the output of movement prediction algorithms. Such algorithms provide the means for pro-active management of resources. The algorithm used takes into account the randomness in the movement of the mobile user as well as the already identified movement patterns to reach combined decisions.

In this paper, we study a quite different cache management scheme: different portions of the original cache are relocated to all the cells being adjacent to the one currently used. A new algorithm for the movement prediction of the user is proposed and its performance is evaluated through Prolog - based simulations. Lastly, based on the simulation results of the path prediction algorithm (PPA) we evaluate the performance of the new cache management scheme. The proposed architecture is named ESW4 (Enhanced Scheme for WWW computing in Wireless communication environments).

The rest of the paper is structured as follows. Section 2 presents the considered architecture for the wireless infrastructure. In the same section we also discuss the "moving" cache approach. Such approach is based on a cache relocation scheme and a PPA. The cache relocation scheme is presented in Section 3, where we also discuss issues related to the performance of proxy caches. Our new PPA is presented in Section 4. In Section 5, we discuss additional details - considerations about the simulation model. In Section 6 we provide the results of the simulation of the PPA. Based on these results we have also simulated the entire ESW4 architecture. The relevant results are also provided in Section 6. We summarize the presented work in Section 7 and identify areas for future research.

2. SYSTEM ARCHITECTURE - THE "MOVING" CACHE APPROACH

Figure 2 provides an illustration of the wireless CPN environment assumed for this study. The environment consists of a number of base stations (BS) interconnected by means of local area networks (e.g., Ethernet, ATM), mobile & fixed hosts. Individual LANs are interconnected by means of routers; one of them is acting as an Internet gateway. Each BS comprises a radio transceiver and a support workstation. The latter deals with all the signaling needed for the roaming of the mobile host. In addition, the BS plays the role of the Support Station encountered in the Mobiscape project.

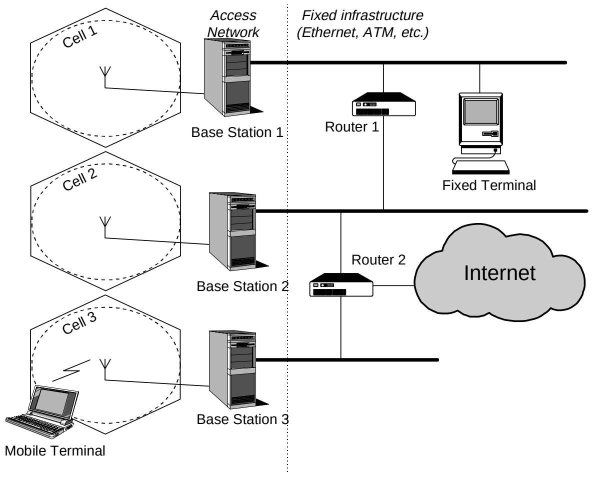


Figure 2. Network architecture

Wireless cells are assumed to be hexagonal and cover the entire surface of the installation (e.g., a university campus, a large office, an industrial facility). BSs maintain information regarding the mobile stations that are currently under their control. In the event of a handover, the involved BSs (of the current and the target cells) need to consult their information base, and collectively undertake specific actions (e.g., reservation/release of resources along the used paths, diversion of connections, ARP[2] updates). In the majority of wireless architectures, user profiles are stored in specialized nodes within the user's home sub-network (the part of the network the user administratively belongs to). When the mobile terminal migrates to a sub-network different from its home, the user profile database (home registry) is queried and the relevant information is forwarded to the visited network by means of specialized inter-network signaling [3].

The home registry of a mobile terminal may also incorporate a PPA (Figure 3). Such algorithm provides, with adequate precision, indications on which cells the terminal is likely to be handed-over if it keeps on roaming. The invocation of the PPA can be performed at some time after the entrance of the mobile terminal in the current cell. From that point, the current BS relocates its cache (or parts of it) to the BSs indicated by the PPA. The mobile terminal is likely to attach to those base stations (also referred to as Target BSs) in the near future. It is not wise to invoke the PPA (and, thus, relocate the accumulated cache) before the above mentioned time period elapses as the cache can still be augmented by user requests and, thus, better hit rates can be achieved. Time statistics held within the home registry could also assist in determining the proper time for PPA invocation (e.g., the algorithm could be invoked right after the recorded mean cell residence time - see Section 5.2).

The sequence of actions undertaken by various network entities as well as the inter-network signaling required for their completion are depicted in the Message Sequence Chart (MSC [27]) of Figure 4.

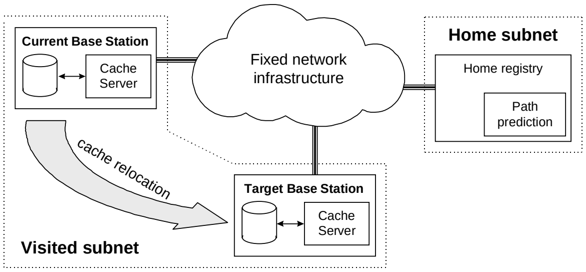


Figure 3. The "moving" cache technique

In Figure 4, the **Determine_Target[MT_ID, BS_ID]** signal is used for triggering the PPA in the home registry. Its parameters denote the identification of the MT for which the algorithm should be executed as well as the identification of the current BS. The Home Registry invokes the PPA and notifies, through the **Relocate_Cache[MT_ID, Target_BSs, HO_Probabilities]** signal, the current Base Station. The Target_BSs parameter is a list that contains the identifiers of all the neighboring cells (to the one currently used by the terminal). The HO_Probabilities parameter is also a list which assigns probability values to the items of the Target_BSs list. Practically, such probabilities denote the likelihood of MT's entrance in the respective cells. The current BS, relocates its cache, or parts of it, to the neighboring BSs according to the probabilities found in the HO_Probabilities list.

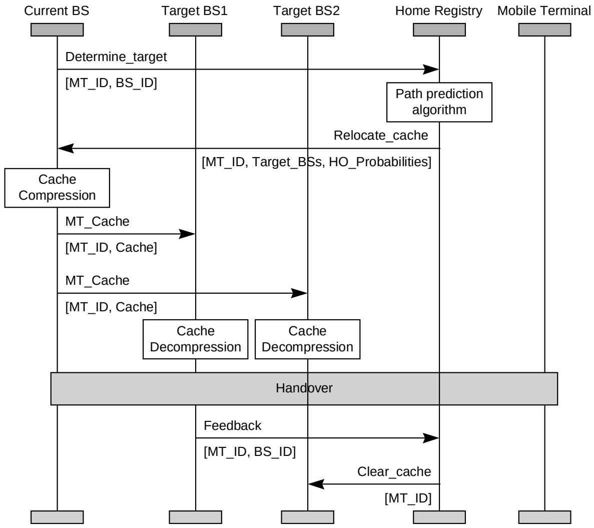


Figure 4. Message Sequence Chart for cache relocation and handover

The relocation of the cache is realized by means of the MT_Cache signals shown in Figure 4. The cache segments contained is those signals have been previously compressed by the current BS. The information is de-compressed by the target BSs and fed into their local cache. The overall procedure is executed in parallel to MT's roaming, without obstructing its communication through the current BS.

At some time after the relocation of its cache, the MT executes a handoff operation. The BS to which the MT has actually been handed over notifies the Home Registry through the **Feedback[MT_ID, BS_ID]**. Such feedback signal is essential for the operation of the PPA. It could also be enriched by a parameter indicating the actual time spent by the user in the previous cell which is essential for the timely invocation of the PPA. The Home Registry that has complete knowledge of which BSs have received MT_Cache signals, notifies them, through the **Clear_Cache[MT_ID]** message. Clear_Cache triggers the removal from BS caches of those items pertaining to the designated MT.

3. CACHE RELOCATION SCHEME - CACHE PERFORMANCE ISSUES

In the following paragraphs we investigate the performance of a cache relocation scheme which involves transmitting the entire (100%) cache from the current BS to the BS indicated by the PPA (the algorithm output). Furthermore, to deal with the cases of path prediction misses, the scheme involves the transmission of the most popular segment (a 70% percentage) of the accumulated cache to two of the other neighboring cells (the second best

[2] Address Resolution Protocol

predictions of the PPA). Even lower percentages (i.e., 30%) are relocated to the remaining neighboring cells. Such segments are referred to as partial caches.

This approach is quite similar to the Shadow Cluster [29] technique proposed for wireless ATM LANs. In the Shadow Cluster scheme, bandwidth is provisionally reserved in the cells adjacent to the one currently used by the roaming mobile terminal. Relocating the entire cache to all the adjacent cells is not a sound strategy, similarly to the bandwidth problem studied in [29], as it may lead to non-optimum use of limited resources in the involved base stations (i.e., the disk or bandwidth capacity of the candidate base station may be exceeded and thus, refuse the proxy service to other roaming users or force the release of their connections, respectively). The use of a proxy cache within the access network is aligned with the general architecture suggested by the Wireless Application Protocol (WAP) Forum. WAP uses a proxy/gateway within the access network to convert formats and adapt the exchanged traffic to standards known to the mobile terminal. A quite similar task could be assigned to our proxy server.

We should also note the possibility of retrieving the requested resources, after the occurrence of a handover, from the previous BS. If the two BSs (previous and current) are attached to the same LAN, such an alternative is feasible. The proxy within the current BS may relay incoming requests to the previous BS. If, however, the old and the current BSs are not interconnected through the same LAN (see Figure 2), then the suggested relocation scheme is advantageous for the rapid dispatch of WWW requests.

One of the issues under study in this work is the performance achieved by the caching system of the BS proxies in the event of path prediction misses. In the following paragraphs, we discuss how we have modeled the behavior of caches and partial caches.

We have adopted a cache size of 2 MB per roaming user, which seems a reasonable volume of relocatable data. Based on measurements taken in our laboratory, we assume that a cache size of 2 MB can accomplish a hit rate of 17-20%. Such hit rate was reported by the WinProxy software [41], configured to keep the local cache size under the 2 MB limit and accessed by a single HTTP client. When the WinProxy cache reaches its maximum configured size, a garbage collection (GC) mechanism is activated and the oldest cache objects are deleted first. The GC mechanism brings the size of the cache down to the 85% of its maximum configured size. We have monitored the performance of the proxy system after a time period of 3 weeks (from system's initialization) with regular workload (thus, we allowed the cache system to refine/filter-out the occasionally referenced resources from the really important ones). The adopted cache hit rate accounts for a worst-case scenario (scenario where the cache accumulated for one user encompasses completely different files from that accumulated for another user). Under regular system operation, the caches from two or more users may be mixed and thus, accomplish hit rates higher than the one reported above. Under all conditions though, the cache hit rate cannot exceed the 50% ceiling reported in the WWW proxy literature [1].

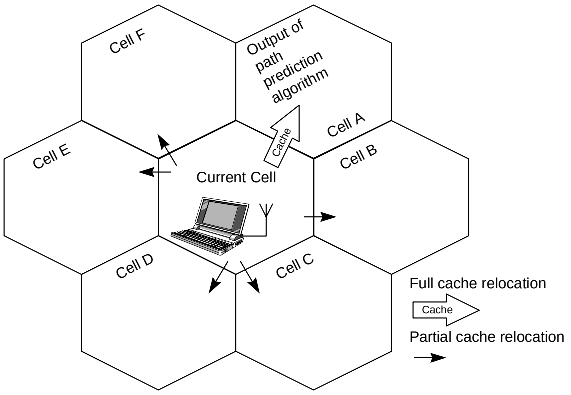


Figure 5. Full and partial cache relocation strategies

The accumulated cache incorporates a set of WWW resources (HTML documents, images, etc.). The size of this set can be roughly estimated by considering the reported distribution for WWW resource sizes and the adopted total cache size of 2 MB. According to [7], the lognormal distribution provides the best fit for Web file sizes less than 133 KB (this family of files accounts for the 93% of the total population of files available throughout the WWW[3]). The parameters of the lognormal distribution have been estimated at the following values: μ=9.357, σ=1.318. The division of the configured maximum size of the proxy cache (i.e., 2 MB) by the mean resource size (i.e., 9.357 KB) yields an approximate mean cache population of 220 items.

The probability of accessing one of the resources stored in a proxy cache can be calculated through the Zipf distribution [21], [16], [7], [2], [4]. Specifically, the number of references to cached item i (**NoR$_{\mathbf{i}}$**) satisfies (1):

$$NoR_i = \frac{a}{rank(i)^Z} \qquad (1)$$

In (1), **rank(i)** denotes the position of cached item i after the sorting of the cache population on the basis of references (i.e., the most popular item is ranked first, the second most popular is ranked second and so forth), **a** is a constant value while **Z** is the Zipf parameter. The Zipf parameter assumes a value close to 1; in [4], Z was estimated at 0.85[4], while in [16] a value of 0.986 was proposed for the parameter. Based on (1), we may induce the conditional probability, P_{hit}, of a cache hit in a partial cache containing the most popular part of the original cache, given a cache hit in the original cache, as follows:

[3] The remaining 7% follows the power-law or Pareto distribution.

[4] This value has been adopted in our measurements - simulations.

$$P_{hit} = \frac{\sum_{i=1}^{\lfloor cache\_size \cdot C \rfloor} NoR_i}{\sum_{i=1}^{cache\_size} NoR_i} = \frac{\sum_{i=1}^{\lfloor cache\_size \cdot C \rfloor} \frac{a}{rank(i)^z}}{\sum_{i=1}^{cache\_size} \frac{a}{rank(i)^z}} = \frac{\sum_{i=1}^{\lfloor cache\_size \cdot C \rfloor} rank(i)^{-z}}{\sum_{i=1}^{cache\_size} rank(i)^{-z}} \qquad (2)$$

In (2), **cache_size** denotes the number of WWW resources (HTML, image files, etc.) found within the proxy cache (i.e., 220). The parameter **C** denotes the percentage of the original cache that was relocated. Note that (2) assumes that P_{hit} is proportional to the total number of references to the items in the partial cache. Figure 6 plots P_{hit} as a function of C.

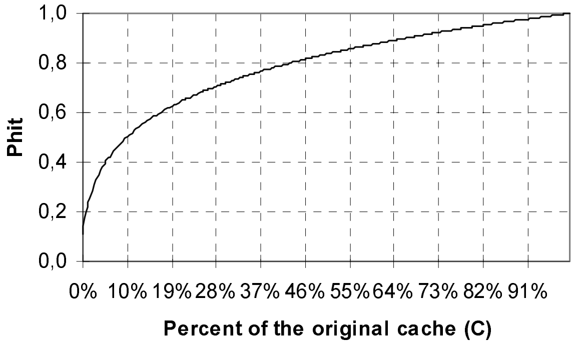


Figure 6. P_{hit} Vs relocation percentage

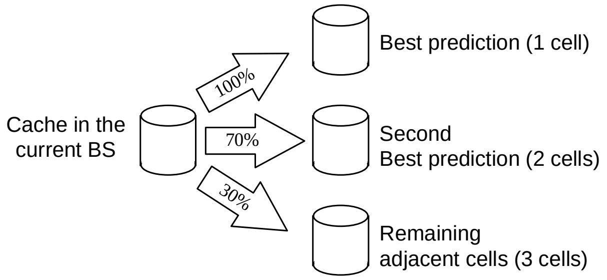

Cache in the current BS
100%
70%
30%
Best prediction (1 cell)
Second Best prediction (2 cells)
Remaining adjacent cells (3 cells)

Figure 7. Cache relocation percentages

Figure 6 shows that a relocation of the 70% of the original cache accomplishes a P_{hit} of 90%, while a relocation of the 30% gives a P_{hit} close to 70%. In our simulation model, we have adopted those two values for the relocation percentages to the second best selections of the prediction algorithm and the remaining neighboring cells respectively (i.e., a total of 5 adjacent cells receive partial caches - we exclude the best selection of the prediction algorithm to which we relocate the 100% of the original cache). This approach is illustrated in Figure 7.

Let P_{new} denote the unconditional probability of a hit in the partial cache. We have:

$$P_{new} = P_{old} \cdot P_{hit} \qquad (3)$$

where P_{old} is the probability of a hit in the original cache. Here we assume that P_{old}=0.2, consistently with our measured hit rate of 20%.

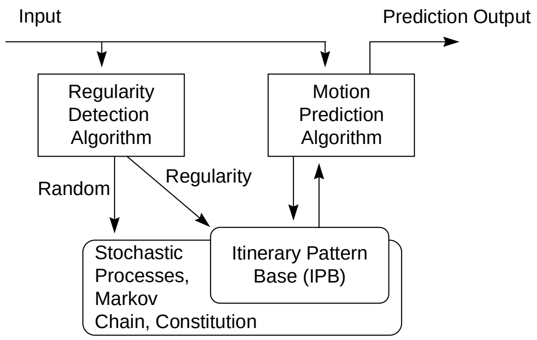
P_{new} is a measure of the efficiency of the partial cache right after its relocation in a neighboring base station. In the event of cache misses (i.e., the designated item could not be found in the partial cache), the numerator (2) changes and P_{new} needs to be updated. The denominator in (2) remains unchanged as it is only dependent on the total allowable set of cached items (i.e., the cache size) which, in our case, is kept constant. If, at some point t in time, a reference to a resource, Req(t), caused a cache miss (i.e., the requested resource was not found in the ⌊cache_size·C⌋+k items of the cache, where k is the number of resources previously fetched in the proxy cache as a result of other misses), or a cache hit, P_{new} is updated as shown in (4). Equation (4) shows, as expected, that P_{new} increases to P_{old}, as the proxy cache in the new base station fills up.

$$P_{new} = \begin{cases} P_{old} \cdot \dfrac{\sum_{i=1}^{\lfloor cache\_size \cdot C \rfloor + k} rank(i)^{-z} + (k+1)^{-z}}{\sum_{i=1}^{cache\_size} rank(i)^{-z}}, \text{ if Req(t) is a miss} \\ \text{unchanged, if Req(t) is a hit} \end{cases} \qquad (4)$$

4. PATH PREDICTION ALGORITHM

As mentioned above, the use of a PPA in a mobile/wireless network architecture allows the optimal use of limited network resources such as disk capacity or bandwidth in base stations. A number of PPAs have been recently proposed in the literature of wireless data networks. Notable examples of such work are the algorithm proposed by Liu et al. [34] and the Liu - Maguire algorithm [31], [32]. More recent works on this area include the Mobility Estimation scheme proposed in [13], for bandwidth reservation in QoS-aware cellular communication environments, and the effort to expedite the location paging process through mobility prediction as described in [9].

The work presented in [34] uses pattern matching techniques and Extended, Self Learning, Kalman filters to estimate the future location of mobile terminals and, thus, perform advance resource reservation and optimal route establishment in ATM based architectures. User Mobility Patterns (UMB) are stored in a database and fed to an approximate pattern matching algorithm to allow estimation (Global Prediction, GP) of a terminal's inter-cell movement direction (deterministic model). The Kalman estimator deals with the randomness in user movement by tracking intra-cell trajectory (stochastic model - Local Prediction, LP). The two models are combined together (Hierarchical Location Prediction) for the derivation of a semi-random movement trajectory. Simulation of the algorithm has shown that it accomplishes a high degree of prediction accuracy as soon as the Kalman filter becomes stable.


Input
Prediction Output
Regularity Detection Algorithm
Motion Prediction Algorithm
Random
Regularity
Stochastic Processes, Markov Chain, Constitution
Itinerary Pattern Base (IPB)

Source: Liu - Maguire, 1995

Figure 8. Predictive Mobility Management algorithms

The Liu-Maguire algorithm is based on Mobile Motion Prediction (MMP) scheme for the prediction of the future location of a roaming user according to his movement history patterns. MMP is "based on the fact that everyone has some degree of regularity in his/her movement, that is, the movement of people consists of random movement and regular movement and the majority of mobile users has some regular daily (hourly, weekly, ...) movement patterns and follow these patterns more or less every day ...". The scheme consists of Regularity-Pattern Detection (RPD) algorithms and Motion Prediction Algorithm (MPA). Regularity Detection is used to detect specific patterns of user movement from a properly structured database (IPB: Itinerary Pattern Base). Three classes of matching schemes are used for the detection of patterns namely the state matching, the velocity or time-matching and the frequency matching. The Prediction Algorithm (MPA) is invoked for combining regularity information with stochastic information (and constitutional constraints) and thus, reach a decision - prediction for the future location (or locations) of the terminal. Figure 8 provides an overview of the suggested scheme. Simulations of the proposed scheme have shown a maximum prediction efficiency of 90 - 95%. The authors have adopted the Liu-Maguire algorithm in the architecture presented in [22].

The Mobility Estimation scheme discussed in [13] uses an aggregate history of observations for achieving estimation of the directions and hand-off times of active connections in adjacent cells.

In this paper, we do not adopt any of the PPAs discussed above. Instead we propose a new algorithm, which is based on well-established Artificial Intelligence (AI) techniques for machine learning. More specifically, we adopt the use of a learning automaton [38]. In the past, there were also other proposals to consolidate AI techniques (e.g., genetic algorithms) with the technology of mobile computing/communications for predicting terminal movement [33].

Learning automata are finite state adaptive systems that interact continuously in an iterative fashion with a general environment. Through a probabilistic, trial-and-error response process they learn to choose or adapt to a behavior which generates the best response. In the first step of the learning process, an input is provided to the automaton from the environment. This input triggers one of a finite number of candidate responses from the automaton. The environment receives and evaluates the response and then provides feedback to the automaton. Such feedback is used by the automaton to alter its stimulus-response mapping structure to improve its behavior.

Learning automata are generally considered as robust but not very efficient learners. They are relatively easy to implement. Generally, the operation of the learning automaton is based on a state transition matrix, which contains the one-step transition probabilities P_{ij} from the current state i to the next state j. Different approaches have been proposed for the updating of the state transition matrix after the reception of environment feedback. In this paper we adopt the behavior of a Linear Reward-Penalty ($L_{R\text{-}P}$) Scheme. When the automaton selects the right response, the positive feedback received by the environment causes the respective state transition to be "rewarded" (i.e., its probability is increased by some pre-arranged step) while the probabilities of the state transitions that were not selected (remaining transitions from the same state) are "penalized" (decreased) uniformly to keep the probability sum to 1. If the proposed response is not appropriate (i.e., a negative feedback was received from environment) a reverse approach is followed; the probability value of the selected transition is "penalized", while the remaining transitions are evenly rewarded to balance the decrease. This behavior is shown in (5):

$$\text{Transition } (\mathrm{i} \rightarrow \mathrm{j}) \text{ received positive feedback: } \begin{cases} P_{ij} = P_{ij} + w(1 - P_{ij}) \\ P_{ik} = P_{ik} \cdot (1 - w),\ k \neq j \end{cases} \tag{5}$$

$$\text{Transition } (\mathrm{i} \rightarrow \mathrm{j}) \text{ received negative feedback: } \begin{cases} P_{ij} = P_{ij} - w'(1 - P_{ij}) \\ P_{ik} = P_{ik} \cdot (1 + w'),\ k \neq j \end{cases}$$

In (5), *w* denotes the rewarding step while *w´* denotes the penalizing step. Those two values may be equal (a successful decision is equally important than an unsuccessful decision) or different (success is more important). In our implementation/simulation, both *w* and *w´* assumed different, but constant, values (see Section 6.1).

Upon invocation, the automaton selects, as candidate future state, the state with the highest probability. After consecutive interactions with the environment, some state transitions will have probabilities close to 1 (which will remain stable) while others will have near-zero values (automaton convergence).

PreviousCell_ID	CurrentCell_ID	FutureCell_ID	TimeSlot	ProbabilityValue	TimeStamp

Figure 9. Layout of state transition matrix entries

In our prediction algorithm, the learning automaton operates on top of an itinerary database each record of which has the layout shown in Figure 9 for some specific user. The set of entries in the itinerary database which are devoted to a single user, in fact, perform the mapping (Previous Cell, Current Cell, Future Cell, Time Slot) → Probability Value.

In ESW4, time is assumed divided in slots of 15 minutes (e.g., only values like 10:00, 10:15, 10:30, 10:45 can be found). Whenever a prediction request arrives at the home registry of the mobile terminal the entries pertaining to the chronologically closest time slot are taken into account provided that the distance to them does not exceed the 1 hour limit (i.e., 4 time slots). If no appropriate state transitions were found then new entries are introduced in the matrix. The *TimeStamp* field indicates when was the particular record consulted for the last time. When the state space exceeds a pre-configured storage allotment a garbage collection procedure is initiated and the oldest entries are deleted from the matrix[5]. This feature, in addition to the adopted partitioning of time, helps in keeping the state transition matrix at a reasonable size.

For a large set of cells/base stations the above described matrix can be partitioned into a number of sub-matrices each one pertaining to a particular region (e.g., a building, a cluster of geographically collocated base stations). In such case, reference pointers are envisaged in the entries of the sub-matrices to direct the queries to peer processes which implement the learning automaton (by applying to other sub-matrices) in other areas of the broader infrastructure. The learning automaton is collectively implemented by such processes dispersed throughout the CPN.

If the automaton decision is correct (positive feedback) then the matrix entry that contributed to this decision (with the chronologically closest time slot) is rewarded while the probability values of the remaining entries are reduced (penalized) as already discussed. The reward/penalty procedure applies to all the entries of the matrix that refer to the same tuple of time slot, previous and current cell (the probabilities of all these entries should sum up to 1).

When a mobile terminal powers-up in a new cell (not previously visited by the same terminal) a number of records are automatically inserted in the itinerary database. These records contain equal probability values (i.e., 1/6 for cells assumed hexagonal in shape), the same current cell identification (CurrentCell_ID) and refer to the same time slot. The identification numbers of future (FutureCell_ID) and previous (PreviousCell_ID) cells reflect all the adjacent cells.

5. SIMULATION MODEL DETAILS

To evaluate the effectiveness of the suggested cache relocation technique we simulated the movement of a nomadic WWW user throughout a wireless CPN infrastructure. Below we discuss two of the most important issues taken into account for the simulation model, namely the modeling of the traffic generated by the WWW user as well as cell residency times.

5.1 WWW traffic modeling

In our simulation model, the behavior of the nomadic WWW user was compliant with the traffic models and statistics reported within the WWW research community over the past years. Specifically, the random variable X used to model the transfer time for Web resources (documents, images, etc.) follows the Pareto (also known as power law) distribution [14], [15]:

$$F(x) = P[X \le x] = \begin{cases} 1-(\dfrac{v}{x})^m, & v \le x \\ 0, & x < v \end{cases} \qquad (6)$$

In (6), the shape parameter m has been estimated in the range 1.0-1.3 [14]. For the simulation model, we have adopted m=1.2. As lower bound, v, for the random variable X we have used the value of 1.

In [14], it is shown that WWW traffic exhibits characteristics consistent with self-similar traffic models. Self similarity is attributed to the multiplexing of a large number of ON/OFF sources where both the ON and the OFF period lengths are heavy tailed processes. ON times correspond to WWW resource transmissions while the OFF times correspond to intervals of client/browser inactivity. Furthermore, OFF times are classified either as **Active** (attributed to client processing delays: document parsing, resource rendering) or as **Inactive** (attributed to user think time). Active OFF times are in the range of 1 msec - 1 sec. Since the time granularity in our simulation environment is in the order of 1 sec, we model the client processing delay as a 1 sec constant. In [7], the Weibull distribution [42] is used to model the Active OFF times. Inactive OFF times are modeled through the Pareto distribution with m=1.5 and v=1, [7].

A pictorial representation of the above assumptions is provided in Figure 10. As shown in Figure 10, the times needed for the transfer of additional resources (e.g., GIF, JPEG images, etc.), the need for which is identified after parsing, can be overlapped (this is the approach followed by most contemporary multi-threaded WWW clients). In [7], the number of embedded references per document is calculated by applying thresholds to the Weibull distributed Active OFF times (i.e., if the Weibull distributed OFF time is less than the threshold value then it is assumed that an embedded reference has been encountered and the retrieval of the appropriate resource has been initiated). Moreover, in the same paper, the distribution of the count of embedded references (more specifically, the logarithm of this figure) has been shown to follow the Pareto distribution (with parameters v=1 and m=2.43). Based on the assumption that the transmission of embedded references can be overlapped (Figure 10), additional resource transfer is invoked as a result of a Bernoulli trial with parameter q. The Bernoulli parameter q can be calculated from the Weibull distribution as the probability of exceeding the 1 sec threshold of Active OFF times (this event can be interpreted as the absence of embedded references in the document):

$$q = P(\text{OFF time} > 1) = 1-(1-e^{-(\frac{1}{\text{a}})^{\text{b}}}) = e^{-(\frac{1}{\text{a}})^{\text{b}}} \qquad (7)$$

[5] Provision is taken to keep the probability values consistent after the deletion of some entries.

Equation (7), for a=1.46 and b=0.382 (values estimated in [7]) yields q=0.421. This figure is very close to the WWW statistics reported in [11]. On the basis of the above, the ON/OFF behavior of a WWW traffic source can be modeled through the chain shown in Figure 11.

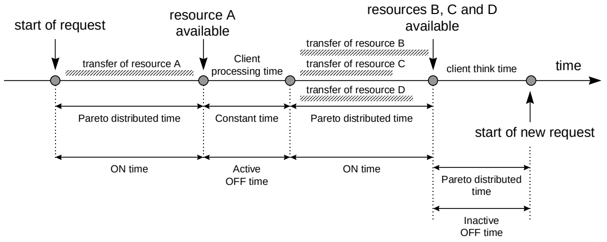


Figure 10. Time distributions for WWW browsing

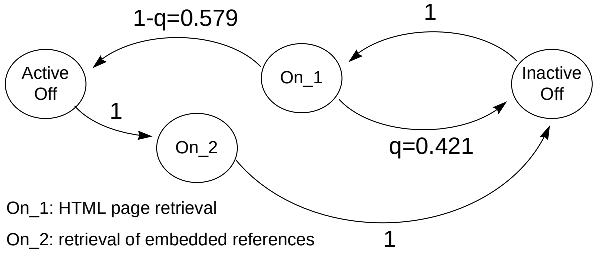

1-q=0.579
1
Active Off
On_1
Inactive Off
1
On_2
q=0.421
On_1: HTML page retrieval
On_2: retrieval of embedded references
1

Figure 11. WWW traffic chain model

Lastly, we should point out that the resource transmission times are not somehow correlated with resource sizes (i.e., file sizes). A large file may be retrieved by a site which is speedily accessed while a small file may be fetched by a remote location with considerable time delays.

5.2 Cell residence times

Another key problem that we had to resolve for completing the simulation environment was the modeling of the time spent by the nomadic user within the current cell (also referred to as residence time). We had to adopt a realistic, yet simple model. The adoption of exponentially distributed residence times is faced with skepticism as the distribution does not exclude values close to 0 (irrespectively of how high the mean value is); instead, it shows particular "preference" to them. In general, such values are undesirable in our modeling. One alternative to the exponential distribution scenario is to cut-off generated time values found below a certain threshold. A mean time value can be determined by dividing an average micro-cell diameter (50m) by a typical walking velocity (2 km/h). This yields a handover rate (the parameter of the exponential distribution) of 0.011 handovers/sec. The cut-off threshold has been set to the value of 10 sec (Figure 12).

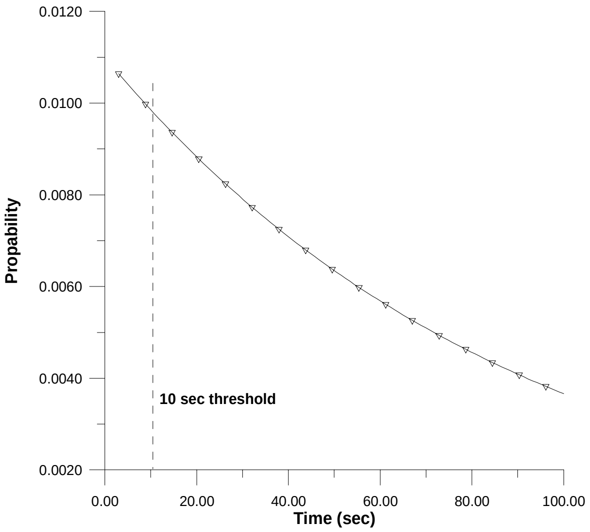


Figure 12. Distribution for cell residence times

6. SIMULATION RESULTS

6.1 Simulation of the path prediction algorithm

We have programmed the logic of the automaton-based PPA in Prolog, a 5th generation (declarative) programming language (5GL) widely used in AI applications [10]. Specifically, we have made use of the Arity/Prolog interpreter environment [5]. The adoption of Prolog has enabled the very rapid prototyping of the automaton due to the easy implementation of a database/state transition matrix structure similar to one discussed before. Additionally, the declarative character of the language renders searching within this database (also referred to as knowledge base) an easy task.

We have logged the behavior of the automaton throughout a period of 7 days and a total number of 592 handovers. The rewarding step, *w* (see equation 5), assumed a value of 0.1 while the penalizing step, *w´*, assumed a value of 0.02. Such values were arbitrarily chosen. The performance achieved by the PPA was not tested against other values of the rewarding/penalizing steps. The sensitivity of the PPA to changes in the rewarding/penalizing steps is an open issue that deserves more study.

The automaton was applied to the movement of one of the authors within the building of our Department. The movement patterns which were fed to the automaton were quite similar in terms of cell border crossings (i.e., the movement from room X to room Y always followed the same route within the building). The movement patterns were fed to the automaton with considerable time variance (i.e., similar itineraries were fed to the automaton in various - not exactly random - time instances of the considered days).

As shown in Figure 13, which plots the performance achieved by the automaton, low hit rates were logged during the first 3 days of the algorithm execution. During those days, due to the absence of database information, the automaton constantly populated the transition matrix (see Figure 14). Since no relevant entries were found in the knowledge base, the algorithm hits logged during those days were mainly random selections (with probability 1/6 due to the cell adjacency/placement). In certain cases, this random selection is also influenced by the cell previously visited by the roaming terminal (in the absence of other information, we assume a linear movement of the terminal; therefore, the automaton, instead of making a completely random selection, points as candidate cell the symmetrical cell to the one previously visited). This strategy is not applied if some of the involved probabilities is greater than 1/6 (i.e., at least once, the automaton has received environment feedback).

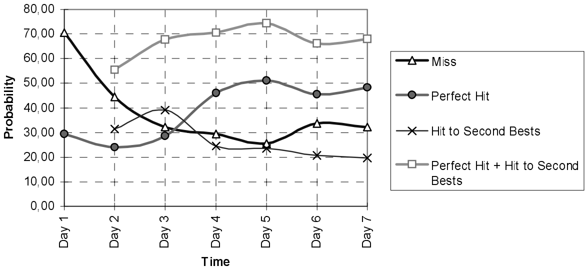


Figure 13. Path prediction efficiency

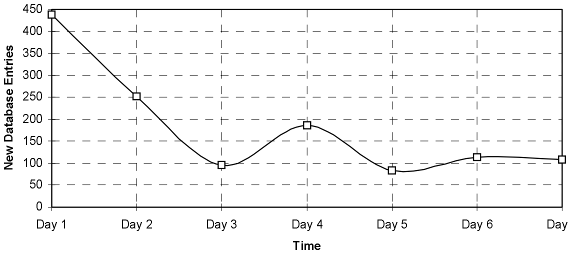


Figure 14. New entries in the automaton database

In Figure 13, we have plotted the probability that the main algorithm output is the one to which the terminal has really been handed over ("Perfect Hit" series/line). We have also plotted the probability that some cell, classified by the automaton as the second best prediction (to the one which has been selected) is the cell to which the terminal has actually been handed over (such probability is reflected by the "Hit to Second Bests" series/line). We have performed this classification for two additional cells (i.e., the automaton output consisted of the three most probable future cells). The logging of this probability was necessary since the proposed algorithm is largely based on the partial relocation of the accumulated cache to certain neighboring cells (i.e., 70% to the "Second Best" predictions). In principle, we show that a relatively simple prediction algorithm can achieve quite similar performance with the more complex (and difficult to implement) schemes, if combined with an approach like the Shadow Cluster principle.

In terms of comparison, the simulation of the Liu - Maguire algorithm, for a medium randomness factor (~ 50%) exhibits a performance quite close to 50% for a simulation period of 5 weeks (without the presence of constitutional constraints). This is the performance that our learning automaton achieves only after a simulation period of 1 week. If the randomness factor drops to the 0% lower bound, the efficiency of the Liu - Maguire algorithm (for the same simulation period) is in the range 90 - 95%.

In [34], it is shown that the performance of the prediction algorithm is quite close to that achieved by the Liu - Maguire algorithm when the randomness factor is fairly low (less than 30%). In higher levels of randomness, prediction is enhanced through the Local Prediction (LP) mechanism (based on the Self-Learning Kalman Filter). In those cases, the performance of the combined algorithm (e.g., Global Prediction - based on pattern matching - and Local Prediction) remains stable at 75%. The Global Prediction algorithm uses a static database of mobility patterns (User Profile) where unclassified/unrecognized patterns are not saved. Table 1 provides an overview of this comparison.

Algorithm/ Criterion	**Liu - Maguire**	**Hierarchical Location Prediction**	**Learning Automaton**
Implementation / Computational complexity	Medium	High	Low
Performance (at medium randomness levels)	~ 50%	~ 75%	~ 68% (= Perfect Hit + Hit to Second Bests)

Table 1: Comparison of Path Prediction Algorithms

6.2 Simulation of the cache relocation scheme

The simulation model for the cache relocation scheme was programmed in MS-Visual C++ ver.5. The most important metrics monitored through the simulation program were

- the average delay (waiting time) perceived by the user in his WWW requests,
- the number of WWW connections that were interrupted by handovers,
- the percentage of interrupted WWW connections (i.e., [handover interrupted connections] / [total number of connections]),
- the number of cache items which were relocated from the current BS to the BS used by the MT after the occurrence of a handover, and,
- the achieved cache hit rate.

The basic parameters adopted for the simulation program are summarized in Table 2.

Parameter	Value
A) Efficiency of the PPA	
• Probability for Perfect Hit	48%
• Probability for Hit to Second Bests	20%
B) Cell residence time: exponential distribution	
• lambda parameter	0.011 handovers/sec
• cut-off threshold	10 sec
C) Web browsing	
• Pareto distribution for resource transmission times: m, shape parameter	1.2
• Pareto distribution for resource transmission times: v, lower bound	1
• probability for embedded references, q (see Figure 11)	0.579
• Pareto distribution for inactive OFF times: m, shape parameter	1.5
• Pareto distribution for inactive OFF times: v, lower bound	1
D) Proxy cache performance	
• Number of items (full, 2MB cache per terminal)	220
• Hit rate (full, 2MB cache)	20%
• Relocation percentage for best prediction (1 cell)	100%
• Relocation percentage for second best predictions (2 cells)	70%
• Relocation percentage for other neighboring cells (3 cells)	30%

Table 2. Basic simulation parameters

We have simulated three scenarios:

- in Scenario 1, the cache relocation scheme described above was employed in the wireless infrastructure.
- in Scenario 2, proxy caches existed in BSs but were not relocated; no prediction algorithm was used
- in Scenario 3, no proxy caching was applied in the various BSs (hence, all requested resources were fetched from remote servers).

For each scenario we performed 5 trials involving 300 handovers each. In Figure 15 we plot the Average Waiting Time per request for all three Scenarios. Scenario 1 achieves a 76.1% of the time consumed in Scenario 3, while Scenario 2 achieves a 89.6%.

In terms of handover interrupted connections, the achieved mean percentage values are presented in Table 3.

Scenario 1	Scenario 2	Scenario 3
3.54%	3.83%	4.2%

Table 3. Mean percentage of HO interrupted connections

The observations from Table 3 are very crucial since the HTTP uses TCP as a reliable transport protocol. Apart from the well-known problems in the interaction of the stateless HTTP (ver.1.0) with TCP [39], Caceres and Iftode in [12] have shown how TCP's congestion control mechanism undermines throughput during handovers. They have quantified the performance degradation in TCP connections caused by MT movement across cell-boundaries. In an overlapping cell scenario, handovers cause the throughput of the TCP connection to decrease by 6%, while in the non-overlapping cell scenario with 0-sec rendez-vous delay[6] throughput drops by 12%. Owing to the stateless character of the HTTP, TCP connections conveying HTTP messages are generally short-lived (their duration equals the time of resource transmission time). Limiting the probability of interruption, by a handover, of a WWW connection will, thus, be beneficial for the performance of the underlying TCP layer.

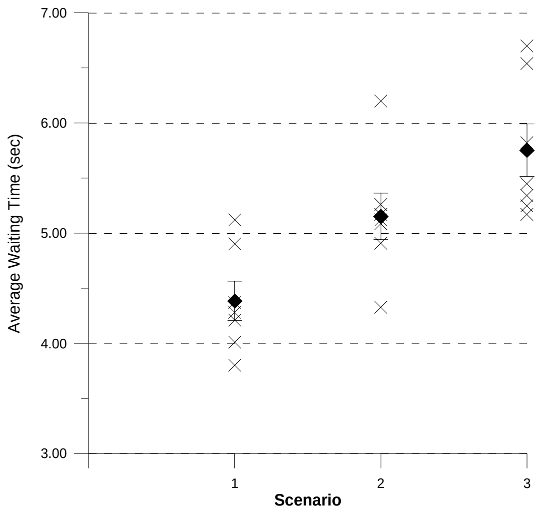


Figure 15. Average Waiting Time for the different Scenarios

Apart from Average Waiting Time and the Percentage of handover interrupted connections, we have also logged, in Scenario 1, the number of items that were relocated in each handover occurrence. This recorded metric shows that in the 17.4% of handover occurrences the relocated volume of data is lower than that allowed by the cache relocation scheme. For example, although the algorithm allows the relocation of 154 items to second best predictions, a lesser number of items existed in the original cache and were eventually relocated. Thus, the utilization factor of the relocation scheme dropped. This utilization factor is increased only if the threshold of the exponentially distributed cell residency times increases. In such case, the cache relocation scheme has the opportunity of recovering from prediction misses since the local cache gradually fills-up to the maximum allowed size even though only a percentage of the original set was received from the previous BS. In terms of proxy cache hit rates, Scenario 1 achieves a 16.2% while Scenario 2 achieves only a 7.63%.

[6] The MT receives control information from the adjacent cell as soon as it leaves the current cell.

7. EPILOGUE

In this paper, we have presented ESW4, an enhanced scheme for the acceleration of WWW browsing in cellular CPN environments. We have briefly discussed some similar architectures (W4). Most of these architectures target to the optimization of the HTTP communication by means of proxies installed in the access network. Provision is not made, though, for the relocation of the proxy cache when the mobile terminal moves to a different BS as a result of a handover.

We have proposed the pro-active relocation of the proxy cache on the basis of the outcome of a PPA. The proposed prediction algorithm, which is based on the well-established learning automaton AI technique, takes into account the time-space regularity in the movement of nomadic users. The algorithm is executed at the access sub-network where the mobile terminal is registered and its implementation is fairly simple. Cache relocation is performed to all the cells adjacent to the one which is currently used similarly to the Shadow Cluster scheme for mobility management. The percentages of the original cache that are relocated are affected by a categorization of cells provided by the PPA.

We have simulated the prediction algorithm which demonstrated a hit rate of approximately 48%. The cache relocation scheme may also benefit from the second best guesses of the algorithm which seem to successfully match the real route of the mobile terminal with a probability of 0.2 (20%). Based on such figures we proceeded with the simulation of the overall scheme. For its simulation we have taken into account the self-similar nature of WWW traffic. The simulation of the proposed algorithm revealed a decrease in the order of 24% in the delay times experienced by the nomadic users.

Our work is well aligned with the research directions in the WAP Forum - W3C co-operation for bandwidth efficiency through the use of “Smart Web Proxies” [25]. A potential improvement of the current work would be the study of more robust learners for use in the PPA. Specifically, we are considering the possibility of employing Fuzzy Set theory for ameliorating the decisions taken by the PPA. Another issues that deserves more study is the sensitivity of the learning automaton used for the PPA in changes of the reward / penalizing steps. Additionally, we are planning to study the efficiency of the PPA in light of constitutional constraints.

8. REFERENCES

[1] M. Abrams, C.R. Standridge, G. Abdulla, S. Williams, and E.A. Fox, “Caching Proxies: Limitations and Potentials”, proceedings of Fourth International WWW Conference, Boston - MA, USA, December, 1995.

[2] C. Aggraval, J. Wolf, and P. Yu, “On Caching Policies for Web Objects”, IBM Research Report RC20619, revision, May, 1997.

[3] B. Akyol, and D. Cox, “Signaling Alternatives in a Wireless ATM Network”, IEEE JSAC, Vol.15, No.1, January 1997.

[4] V. Almeida, A. Bestavros, M. Crovella, and A.de Oliveira, “Characterizing Reference Locality in the WWW”, Technical report BU-CS-96-11, Computer Science Dept., Boston University, 1996.

[5] “The Arity/Prolog Compiler and Interpreter”, Arity Corporation, Concord, MA, 1992.

[6] C. Baquero, V. Fonte, F. Moura, and R. Oliveira, “MobiScape: WWW Browsing under Disconnected and Semi-Connected Operation”, proceedings of First Portuguese WWW National Conference, Braga, Portugal, July 1995.

[7] P. Barford, and M. Crovella, “Generating Representative Web Workloads for Network and Server Performance Evaluation”, proceedings of ACM SIGMETRICS, July, 1998.

[8] T. Berners-Lee, R. Cailliau, A. Luotonen, H.F. Nielsen, and A. Secret, “The World-Wide Web”, CACM, Vol.37, No.8, August 1994.

[9] A. Bhattacharya, and S. K. Das, “LeZi Update: An Information Theoretic Approach to Track Mobile Users In PCS Networks”, proceedings of ACM/IEEE Mobicom ’99, Seattle, USA, August 1999.

[10] I. Bratko, “Prolog Programming for Artificial Intelligence”, Addison-Wesley, 1990.

[11] T. Bray, “Measuring the Web”, Computer Networks and ISDN Systems, Vol. 28, No. 7-11, 1996.

[12] R. Caceres, and L. Iftode, “Improving the Performance of Reliable Transport Protocols in Mobile Computing Environments”, IEEE JSAC, Vol.13, No.5, June 1995.

[13] S., Choi, and K.G., Shin, “Predictive and Adaptive Bandwidth Reservation for Hand-offs in QoS-Sensitive Cellular Networks ”, proceedings of ACM SIGCOMM ‘98, Vancouver, British Columbia, September 1998.

[14] M. Crovella, and A. Bestavros, “Self-Similarity in World Wide Web Traffic: Evidence and Possible Causes”, IEEE/ACM Transactions on Networking, Vol. 5, No. 6, December 1997.

[15] M. Crovella, M. Taqqu, and A. Bestavros, “Heavy-Tailed Probability Distributions in the World Wide Web”, in “A Practical Guide to Heavy Tails - Statistical Techniques and Applications”, R. Adler, R. Feldman, and M. Taqqu (ed.), BIRKHAUSER, 1998.

[16] C. Cunha, A. Bestavros, and M. Crovella, “Characteristics of WWW Client-based Traces”, Technical report BU-CS-95-010, Computer Science Dept., Boston University, July, 1995.

[17] K.Y. Eng, M. Karol, M. Veeraraghavan, E. Ayanoglu, C. Woodworth, and R.A. Valenzuela, “A Wireless Broadband Ad-Hoc ATM Local-Area Network”, ACM/Baltzer Wireless Networks Journal, Vol. 1, May, 1995.

[18] “Radio Equipment and Systems (RES); Digital European Cordless Telecommunications (DECT) - Common Interface - Part 1: Overview”, European Telecommunication Standard (ETS 300 175-1), ETSI, October, 1992.

[19] "Universal Mobile Telecommunication System (UMTS), Service Aspects, Virtual Home Environment, v.2.0.0", UMTS 22.70, ETSI, March, 1998.

[20] R. Floyd, B. Housel, and C. Tait, "Mobile Web Access Using eNetwork Web Express", IEEE Personal Communications, October 1998.

[21] S. Glassman, "A Caching Relay for the World Wide Web", Computer Networks and ISDN Systems, Vol.27, No.2, 1994.

[22] S. Hadjiefthymiades, and L. Merakos, "Improving the Performance of the World Wide Web in Cellular CPN Environments", proceedings of 5th Intl. Workshop on Mobile Multimedia Communication (MoMuc'98), Berlin, Germany, October 1998.

[23] S. Hadjiefthymiades, and L. Merakos, "A Survey of Web Architectures for Wireless Communication Environments", Journal of Universal Computer Science, Springer - Verlag, Vol.5, No.7,1999.

[24] G.A. Halls, "HIPERLAN: the high performance radio local area network standard", Electronics and Communication Engineering Journal, December, 1994.

[25] J. Hjelm, B. Martin, and P. King, "WAP Forum - W3C Cooperation White Paper", Sept., 1998.

[26] B.C. Housel, and D.B. Lindquist, "WebExpress: A system for Optimising Web Browsing in a Wireless Environment", proc. of ACM/IEEE MobiCom '96, New York, USA, October, 1996.

[27] ITU-T Recommendation Z.120, "Message Sequence Chart (MSC)", 1993.

[28] T.F. La Porta, K.K. Sabnani, and R.D. Gitlin, "Challenges for Nomadic Computing: Mobility Management and Wireless Communications", ACM Journal of Nomadic Computing, Vol.1 No.1, 1996.

[29] D. Levine, I. Akyildiz, and M. Naghshineh, "A Resource Estimation and Call Admission Algorithm for Wireless Multimedia Networks Using the Shadow Cluster Concept", IEEE/ACM Transactions on Networking, Vol.5, No.1, February 1997.

[30] M. Leventhal, D. Lewis, and M. Fuchs, "Designing XML Internet Applications", Prentice Hall, 1998.

[31] G.Y. Liu, and G.Q. Maguire, Jr., "A Predictive Mobility Management Algorithm for Wireless Mobile Computing and Communications", proc. of ICUPC '95, Tokyo, Japan, Nov., 1995.

[32] G.Y. Liu, and G.Q. Maguire, Jr., "Efficient Mobility Management for Wireless Data Services", proceedings of IEEE VTC '95, Chicago - Illinois, USA, 1995.

[33] L.Q. Liu, A.T. Munro, and M.H.Barton, "Efficient Mobility Management: A New Flexible Design Algorithm", proceedings of ICUPC '96, Cambridge - MA, USA, September, 1996.

[34] T. Liu, P. Bahl, and I. Chlamtac, "Mobility Modeling, Location Tracking, and Trajectory Prediction in Wireless ATM Networks", IEEE JSAC, Vol.16, No 6, August 1998.

[35] A. Luotonen, and K. Altis, "World-Wide Web Proxies", proceedings of First International WWW Conference, Geneva - Switzerland, May 1994.

[36] J. Mikkonen, J. Aldis, G. Awater, A. Lunn, and D. Hutchison, "The Magic WAND - Functional Overview", IEEE JSAC, Vol.16, No.6, August, 1998.

[37] M. Mouly, and M. Pantet, "The GSM System for Mobile Communications", ISBN: 2-9507190-0-7

[38] K. Narendra, and M. A. L. Thathachar, "Learning Automata: An Introduction", Prentice-Hall, 1989.

[39] V.N. Padmanabhan, and J.C. Mogul, "Improving HTTP Latency", Computer Networks and ISDN Systems Vol.28, 1995.

[40] "The Path towards UMTS - Technologies for the Information Society", UMTS Forum, 1998.

[41] M. Viktora, and M. Rubáš, "WinProxy 1.3, User's Manual", LAN-Projekt, October, 1996.

[42] Y. Viniotis, "Probability and Random Processes for Electrical Engineers", McGraw-Hill, 1998.

[43] "Wireless Application Protocol Architecture Specification", WAP Forum, April, 1998.